EUROPEAN ASSOCIATION FOR MACHINE TRANSLATION

EAMT2018

Proceedings of the

21st Annual Conference of the European Association for Machine Translation

28–30 May 2018

Universitat d'Alacant

Alacant, Spain

Edited by

Juan Antonio Pérez-Ortiz

Felipe Sánchez-Martínez

Miquel Esplà-Gomis

Maja Popović

Celia Rico

André Martins

Joachim Van den Bogaert

Mikel L. Forcada

Organised by

Universitat d'Alacant

Universidad de Alicante

transducens

research group

ISBN: 978-84-09-01901-4

Letting a Neural Network Decide Which Machine Translation System to Use for Black-Box Fuzzy-Match Repair

John E. Ortega
Universitat d'Alacant
E-03071, Alacant, Spain
jeo10@alu.ua.es

Weiyi Lu
New York University, 60 5th Avenue
New York, New York 10011, USA
weiyi.lu@nyu.edu

Adam Meyers
New York University, 60 5th Avenue
New York, New York 10011, USA
meyers@cs.nyu.edu

Kyunghyun Cho
New York University, 60 5th Avenue
New York, New York 10011, USA
kyunghyun.cho@nyu.edu

Abstract

While systems using the Neural Network-based Machine Translation (NMT) paradigm achieve the highest scores on recent shared tasks, phrase-based (PBMT) systems, rule-based (RBMT) systems and other systems may get better results for individual examples. Therefore, combined systems should achieve the best results for MT, particularly if the system combination method can take advantage of the strengths of each paradigm. In this paper, we describe a system that predicts whether a NMT, PBMT or RBMT will get the best Spanish translation result for a particular English sentence in DGT-TM 2016[1]. Then we use fuzzy-match repair (FMR) as a mechanism to show that the combined system outperforms individual systems in a black-box machine translation setting.

1 Introduction

Natural Language Processing (NLP) systems designed to do the same task often belong to different methodological paradigms. At any time in history, the best-scoring systems may tend to come from a particular paradigm. For example, in Machine Translation (MT), the current dominant paradigm is Neural Network-based MT (NMT). The previously dominant paradigm was Phrase Based MT (PBMT), and so on. When comparing MT results for different types of input, systems from certain paradigms perform better on certain types of input and vice versa (Bentivogli et al., 2016). In some cases NMT suffers more than other paradigms (Koehn and Knowles, 2017). Thus, it may be premature to completely abandon "old" methods in favor of "new" ones.

Newer methods, especially NMT, tend to achieve higher BLEU scores than previous methods including PBMT and Rule-based MT (RBMT) systems. However, professional translators and users of computer-assisted translation (CAT) tools seem to prefer PBMT output for particular sentences (Arenas, 2013). Many recent systems (e.g., participants in WMT17 (Bojar et al., 2017)) use NMT, because it obtains higher scoring results, but does not require time-consuming procedures like feature generation or Quality Estimation (QE) to achieve quality MT translations.

CAT tools, and other systems using black-box MT, could benefit from a way of predicting which MT system will perform the best at translating a particular source segment. Such systems which typically use only one MT tool to translate all input could benefit from selectively using the output of multiple systems in this way.

This paper describes a series of experiments that attempt to take advantage of the strengths of alternative systems and combine system output to produce the best result. First, we describe our system, **SelecT**, which uses a neural-network based approach to predict which system provides the best output for translating a particular English sentence to Spanish using: Nematus (Sennrich et al., 2017), an NMT system; Moses (Koehn et al., 2007), a PBMT system; and Apertium (Forcada et al., 2011), an RBMT system. Then we use the MT system predicted to be the *best* [2] to improve pre-

[1] https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory

[2] according to BLEU score

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 209–218
Alacant, Spain, May 2018.

vious work on fuzzy-match repair (FMR), an approach that uses black-box machine translation as its primary method for translating sub-segments to be repaired (Ortega et al., 2014).

Most previous hybrid approaches to MT focus on ways to combine individual translations from different MT systems. In contrast, our system uses multiple predictive model types to choose the optimal sentence-level translations, without previous knowledge of the internal workings of the MT system. **SelecT** predicts which of 3 translation systems will produce the best translation. **SelecT** uses the performance differences seen on various tasks with different data where typically one MT system, be it rule-based, phrase-based, or neural, outperforms the other systems, to improve results by providing sentence-level predictions where often times differences in MT system quality can occur depending on the data.

In a professional setting, MT systems may have a higher cost due to quality performance issues and it would make sense that a translator has the most appropriate translation at hand when using the MT tool. Relying on a single MT system could be costly as shown in previous investigations (Rosti et al., 2007). We propose a prediction system that integrates easily into *any* system that uses black-box MT. Black-box MT systems would use the MT engine that **SelecT** predicts using a pre-translation performance metric. **SelecT** accepts any source sentence input s and produces a translation σ in a transparent way by predicting beforehand the system to use and querying the black-box MT system with the ideal (best-predicted) MT engine to use. Our goal is to measure how well mainstream MT systems perform and compare their differences for commercial use situations where often times translation quality should be determined beforehand to determine economic value.

2 Related Work

2.1 Fuzzy-Match Repair

Our system tests our MT results on an active implementation of fuzzy-match repair (Ortega et al., 2016) that uses Apertium (Forcada et al., 2011) as its MT engine. While previous work (Knowles et al., 2018) has already tested the black-box nature of FMR using 3 MT systems (Apertium, Moses, and Nematus), they do not attempt to predict which of those systems would perform best in a black-box translation task like we do here.

2.2 System Combination

There have been many papers about system combination in MT, so we will only highlight a few of them. Most researchers chose to combine systems using different methodologies. Published in 1994, Frederking et. al (1994) describe a Spanish to English system for synthesizing single translations of each sentence from parts of the translations produced by 3 MT engines: knowledge-based MT (PanGloss), example-based (EBMT) and a lexical-transfer+morphology system. Their combined system scores are measured by the number of keystrokes required to correct the automatic translations. In a similar way, Sańchez-Cartagena et. al (2016) show that an ensemble of an NMT and a PBMT system outperforms each of these systems individually when translating Finnish to English, as measured by BLEU. They use CMU's Multi-Engine Machine Translation (MEMT) Scheme (Heafield and Lavie, 2010) for system combination. MEMT aligns translations using METEOR (Lavie and Agarwal, 2007) and uses a beam search and a variety of features. Chaterjee et. al (2016) describes an MT system called "Primary" that includes an RNN implementation along with Moses. Their work, like others from WMT16 (Bojar et al., 2016), is mainly focused on translation tasks and improving translation by interchanging models. They do not chose the best system for each translation output; rather, they combine systems to produce the best output possible. Unlike approaches of system combination described above, our work focuses on predictions at the black-box, system-level input by predicting, beforehand, the optimum MT system to use. Our models are trained using a minimal amount of features and use sentence-level BLEU scores as the determining metric for labeling positive translation examples.

2.3 Evaluation and Quality Assessment

MT evaluation has been performed using many different metrics, e.g., those described in White et. al (1995). Those evaluations are very helpful to determine which MT system one would use for a specific metric. However, those metrics leave the guesswork up to the MT system or CAT tool user and do not attempt to predict which system to use.

In order to properly combine system output, it is necessary to assess the quality of that output. Formal evaluation requires human intervention (hu-

man translations or evaluations). In contrast, Quality Estimation (QE) (Specia et al., 2013), is a popular paradigm for automating assessment. QE uses a model to predict the quality of a translation without human intervention. The features that are used in QE are typically corpus-level features and are not based on previous (conflicting) translations from a different MT system. Nonetheless, one could add features to a QE system to perform work similar to ours - we skip the QE step for now as we are focused more on measuring how well a particular MT (or combination of MT) system(s) perform. Others have also performed research by measuring system output to determine the best model to use. Nomoto (2004), for example, use a voted language model based on support vector regression to determine a confidence score of a sentence in the translation output and use the highest scoring sentence as the final output. His approach is similar to ours; but, we use a different mechanism for selecting output based on several models to predict sentence-level quality before translating.

3 Methodology

Our work uses a predictive classifier to determine the best MT system for translation when used as a black box such that no prior knowledge of the internal workings of the MT system is necessary. It will allow any system with the ability to call a $translate()$ method access to sentence-level quality without the use of more complex paradigms such as quality estimation.

Combining several MT systems via our black-box method should achieve higher scores than just using one MT system. FMR (Ortega et al., 2016) is a recent example of a black-box $translate()$ method that uses one MT system, Apertium. Their work assumes no dependency on other parts of the MT system. Here, we use the work from Ortega et. al (2016) to show the advantage of having a mechanism to predict the best MT system to use before actually calling the $translate()$ method.

Our work intrinsically compares 3 open-source MT systems using 3 different classifier models: 1) Recurrent Neural Network (RNN), 2) FastText[3] classification, and 3) Logistic Regression (LR) described in section 5.1.1. Each model is created to predict sentence-level quality using BLEU. Then, we use both BLEU and word-error rate (WER) as a performance measurement to determine which model to use in FMR. WER for our experiments is considered as the word-based edit distance between the reference translation and the system translation often called *Levenshtein distance* (Wagner and Fischer, 1974). Our model is somewhat similar to a Quality Estimation model but based on MT engines alone. The prediction model is part of a bigger system that when given a new sentence s and a set of systems: $\{MT_{01}, MT_{02}, MT_{03}, ...\}$ derives a translation by selecting an MT system based on training data. Our hypothesis is that a system that can determine which MT engine to use before actually having the system translation should perform better and offer the best value for the translator or CAT tool user.[4]

After establishing that **SelecT** can select translation engines in a fashion that is beneficial to the user, we evaluate, with WER, **SelecT**'s choices using a system that uses black-box MT, fuzzy-match repair (Ortega et al., 2016). When the FMR system needs to translate any segment, whether an entire sentence or sub-segment of a sentence, it calls upon **SelecT** to determine which engine to use for the source sentence to be translated. Then, FMR calls its $translate()$ method with the MT engine suggested. We test **SelecT** in this paper using: Apertium, Moses, and Nematus. Our experiments measure WER from FMR when using **SelecT**. We aim to improve upon previous results (Ortega et al., 2016) by choosing the best predicted MT system for each translation in FMR.

4 Descriptions of MT Systems

4.1 Apertium

Apertium (Forcada et al., 2011) is a rule-based MT system employing manually created rules and dictionaries for each language pair. It is a community-based MT system that has a lot of contributors and provides an on-line translation tool [5] free for anyone's use. In addition to a large community base, there's a lot of documentation (Forcada et al., 2009) available that explain how the shallow-transfer MT system works.[6] We chose Apertium as the representative rule-based MT system because

[3] `https://github.com/facebookresearch/fastText/`

[4] Users of subscription MT services would only pay for sentences that SelecT chose to translate with a particular system. Thus they would not have to pay more than once for the same sentence.

[5] `http://apertium.org`

[6] Apertium works best with romance language pairs like ES–PT, ES–FR, etc. (Ortega et al., 2016; Knowles et al., 2018)

it's an open-source translation engine already used in a black-box translation system for FMR (Ortega et al., 2016). In order to align experiments with past work, we use the same version (SVN 64348) and language-pair package: apertium-en-es from Ortega et al. (2016). Apertium implements morphology through its modifiable technique called the lt-toolbox. It takes into account language structure by using part-of-speech tagging and chunking.

4.2 Moses

Moses (Koehn et al., 2007) is our representative phrase-based MT system. Previous black-box MT work (Knowles et al., 2018) found that Moses works well as a comparison MT engine.[7] Moses is the most widely adopted (non-neural) open-source *statistical* MT system. It combines statistical models with phrase tables to determine how to precisely translate unseen words. Moses is a complex system that, in our developmental experiments, performs well on word ordering and specific learned punctuation like "<<" and ">>" often used for translating quotation marks in our data. In several cases, Moses was the only MT system to correctly translate rare punctuation marks differences.

As a phrase-based MT system, Moses generally outperforms most other PBMT systems and is generally considered the de facto system to use for open-source MT (Dugast et al., 2007; Schwenk et al., 2012). It has already been compared to various neural MT systems. In particular, work from Junczys-Dowmunt et. al (2016) directly compares Moses against Nematus as does other work (Knowles et al., 2018). For the EN–ES language pair, BLEU scores reported in the work from Junczys-Dowmunt et. al (2016) were similar (about 1.4 difference).

4.3 Nematus

Nematus[8] is a neural MT system from the University of Edinburgh. It is implemented in Python, and based on the Theano framework (Sennrich et al., 2017). One major advantage that is pertinent to this paper is that Nematus uses byte-pair encoding (BPE) which starts from a character-level segmentation and eventually encodes full words as a single symbol (Sennrich et al., 2016). The potential for Nematus to score well on translations that differ at the character-level instead of at the word level is high.

Previous black-box comparison experiments for FMR (Knowles et al., 2018) also use Nematus. In WMT 2016, Nematus outperformed other MT systems with less complexity for feature engineering, i.e., Nematus requires training on word-embeddings alone while other systems, like Moses, require more complex statistical models and configuration parameters.

4.4 Advantages and Disadvantages

Based on the previous work using the 3 MT systems (Apertium, Moses, and Nematus), we believe that **SelecT** should outperform any single system. Each individual MT system has some particular advantage (or disadvantage) that would provide more information to a model for prediction to use when translating an unseen sentence. For example, Apertium may produce quality translations in some cases where morphology or part-of-speech linguistic features are absolutely necessary; Moses may perform better than Apertium on sentences that have frequent phrases; and Nematus will probably outperform the other systems for most sentences. Nematus should also do particularly well on character replacements and other sentences that require one-word deletion or insertion.

Luong et. al (2014) and Alva-Manchego et. al (2017) show that Moses is conservative with deletions, yet good with punctuation. However, both Apertium and Moses are unlikely to do well with lexical complexity (Luong et al., 2014). Apertium is good at making lexical and morphological distinctions. So, while it has been shown to perform worse on English to Spanish language pairs (Ortega et al., 2014), it is still worthwhile to use as a default system for testing due to its expert, handcrafted, methodology that is backed by an (HMM) (Cutting et al., 1992) which is known to classify parts of speech and morphemes well.

Some types of problems that an MT system may find with the test corpus, DGT-TM 2016,[9] relate to the corpus's parliamentary text. It contains punctuation irregularities and a lot of the segments that, due to its legal register, require a one-to-one alignment where the target (Spanish) words should not have to change much despite the language difference (English to Spanish). In addition, the text

[7] We trained Moses on Europarl V7(Koehn, 2005) and tuned it on WMT12.

[8] https://github.com/EdinburghNLP/nematus

[9] https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory

contains several hundred out-of-vocabulary (OOV) words which can be hard to cover with any MT system.

In summary, while Nematus is key to high quality translations, we should not dismiss Apertium or Moses since they translate some segments better than Nematus does.

5 Experimentation

5.1 Settings

Our experiments use several corpora and systems based on previous work on black-box MT (Knowles et al., 2018) and FMR (Ortega et al., 2016). Knowles et. al (2018) does a comparison of fuzzy-match repair using the 3 MT systems described in this paper. For both experiments, we use similar data. First we show that MT systems can be successfully selected; then we use the predictor for fuzzy-match repair. However, since we are trying to reproduce settings similar to (Ortega et al., 2016), there are some changes in the systems used.

5.1.1 MT-Experiments

There are 3 predictive models used to select an MT system based on training data. The implementation of each model is described in further detail below and found on Github [10]. All predictive models used the same DGT-2016 TM [11] for training. We divided DGT-2016 into an 80%/10%/10% split for train/dev/test, respectively. The dev set was used for error analysis and to help better understand the oracle (ensemble) settings. After gathering all of the data for statistical analysis, we used our saved models on the unseen test data. We use the EN–ES language pair from DGT-TM 2016 which contains 203,214 total parallel sentences. We lowercased all sentences and tokenized them using the tokenizer from the Moses baseline run.[12]

We test our predictive models on 3 MT systems (Apertium, Moses, and Nematus). The MT systems were similar in nature as far as the corpora used to train them, although, Apertium doesn't actually require training - it's a rule-based MT system. Apertium is a specific EN–ES version (SVN 64348). Our version of Moses mirrors the baseline[13] except for the training corpus, we train Moses using the EN–ES from EUROPARL v7 (Koehn, 2005) and tune, as in the baseline, on WMT12[14]. Our Nematus MT system is trained on Europarl v7 and News Commentary v10 data[15] (WMT13 training data for EN–ES).

Training is done where the best scoring system (according to BLEU) wins. There are 162571 sentences in the training set. In the training set, Apertium scores best on 26426 sentences; Moses scores best on 54372 sentences; and Nematus scores best on 81773 sentences. For our final test set, a perfect score for the **SelecT** system would be: 3441, 6602, and 10278, respectively. Therefore, we are training on what can be considered the "ensemble" system. Final test results report 2 metrics: 1)BLEU and 2) word-error rate (WER). We use 3 different algorithmic models for training:

1. **Bi-Directional Recurrent Neural Network** In the text we refer to this model as *RNN*. The model uses word embeddings created by Gensim[16] from the DGT-TM 2016 corpus with embedding dimensions of 300. Sentences of more than 100 words in length are discarded. The model itself is created using Theano[17] and has a gated recurrent unit (GRU) (Cho et al., 2014) with 300 hidden units as the recurrent neural layer. We use a dropout rate of 0.5 and RELU (Nair and Hinton, 2010) activation. This model is used with hopes that it has the ability to learn spontaneous words and activate clearly for system label classification where other (non-neural) models would not.

2. **FastText Supervised Learner** We chose the FastText [18] supervised model because it is a quick and efficient model that classifies text. For training, we use 25 epochs. For word embeddings we used a 300 dimension vector. The implementation is very straightforward and our command line options are passed such that the n-gram length is 5. All of our labels were passed in-line following the FastText installation instructions.

   For comparison purposes, FastText could be thought of as a neural net with a single hidden layer using bag-of-n-grams representation (we use 5-grams). This is a generaliza-

[10] https://github.com/AdamMeyers/Web-of-Law/EAMT2018
[11] https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory
[12] http://www.statmt.org/moses/?n=moses.baseline
[13] http://www.statmt.org/moses/?n=Moses.Baseline
[14] http://www.statmt.org/wmt12/dev.tgz
[15] http://www.casmacat.eu/corpus/news-commentary.html
[16] https://radimrehurek.com/gensim/
[17] http://deeplearning.net/software/theano/
[18] `https://github.com/facebookresearch/fastText/`

tion of bag-of-word logistic regression. For classification purposes, FastText works better in terms of classification. Our results show, however, that better classification accuracy does not necessarily result in better translation quality (BLEU).

3. **Logistic Regression** For our Logistic Regression (LR) model we used the popular Python machine learning framework SciKit-Learn v0.19.1 [19]. For sentence representations, SciKit-Learn is used to get bag-of-words (BOW) features and scored via term frequency inverse document frequency (TF-IDF) scores (Salton and Buckley, 1988).

Model training time differs for the 3 models. FastText and logistic regression (generating a bag-of-words representation and features based on TF-IDF features) can both be trained within several minutes (on 12 cores of an Intel Xeon E-2690v2 3.0GHz CPU), while it takes roughly 16 minutes to train the bi-directional recurrent neural network model per epoch (on one NVIDIA P40 GPU). For our purposes during the development stage, the best accuracy for the RNN was observed at 40 epochs. Clearly, in our experiments, the FastText and logistic regression models train faster than the RNN - one may want to consider these times for replication of our work in the future.

5.1.2 FMR Experiments

In order to replicate experiments from Ortega et. al (2016), we use exactly the same settings as they did. They use 1993 test sentences along with a translation memory extracted from DGT-TM 2015. We use an Apertium MT system(Forcada et al., 2011) (SVN 64348) similar to theirs (Ortega et al., 2016).

The other 2 MT systems that are used are Moses and Nematus. For the FMR experiments, we use the MT systems from section 5.1.1 to test on. All 3 systems (Apertium, Moses, and Nematus) make up part of the **SelecT** system that FMR uses when calling its black-box translate method such that the following steps occur:

1. a new source side sub-segment (σ or σ') is proposed for translation from FMR (for more details on FMR consult (Ortega et al., 2014)).
2. σ or σ' is passed as a new sentence to be classified to the **SelecT** system (**SelecT** does not actually run inside of FMR nor does it have knowledge of the internal workings of FMR).
3. the best performing model from previous experiments on **SelecT** (in our experiments it's the FastText model) is used to select whether Apertium, Moses, or Nematus is used to translate the sentence.
4. the black-box component of FMR translates σ or σ' using the selected MT system.
5. the black-box component of FMR returns a new translation τ or τ' respectively.

We use the best performing model (FastText) from our MT experiments to test FMR by allowing it to choose the best MT system when presented a new segment (or sub-segment) from the FMR's *translate()* method call. Results are reported for **SelecT** by measuring the WER produced when using the selected MT systems per sentence.

All systems WER are reported separately and with and without predictive tactics. It is worthwhile to note that there are cases when a fuzzy-match score is not met and the entire sentence (s' from (Ortega et al., 2016)) is translated. In those cases, we *also* use our predictive models from the **SelecT** system to choose an MT system to translate the entire sentence.

6 Results

We provide results of 2 experiments: experiment 1 measures the accuracy of the predictive models in **SelecT** using BLEU and WER as evaluation metrics. Experiment 2 uses **SelecT** as an agnostic predictor to choose an MT system for FMR. For experiment 1, we use 20321 sentences to test the 3 MT systems (Apertium, Moses, and Nematus) with 3 types of classification (RNN, FastText, and Logistic Regression). Table 1 shows how well each system performs in isolation – if we were to use the respective system as the sole translation engine for all 20321 sentences. Table 2 provides provides counts of sentences such that the corresponding model correctly predicts the highest BLEU score. It allows us to review the scores for each of the MT systems (Apertium, Moses, Nematus) at a localized level to show how well each system performs when it out-performs the other systems. For example, using the FastText system as

[19] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

a predictor, Apertium outperforms Moses on 2283 of the 20321 total sentences.

System	BLEU	WER	Unique Tokens
Apertium	20.96	59.91	16773
Moses	30.05	54.02	21711
Nematus	37.36	51.77	26372

Table 1: BLEU, WER, and unique tokens for 3 MT systems

System	RNN	FT	LR	Ref
Apertium	2855	2283	1798	3441
Moses	6530	6553	5983	6602
Nematus	10936	11485	12540	10278

Table 2: Count of sentences for 3 predictive models

For even more details about our predictive models, we present the accuracy of our models in isolation on the 20321 test sentences. Table 3 shows how accurate each model is in predicting the MT system that would perform best using BLEU as the scoring metric. For example, the RNN **SelecT** system predicted the best MT system to use about 66% of the time.

System	Prec.	Rec.	F1	Acc.
RNN SelecT MT System				
Apertium	61.05	50.65	55.37	65.79%
Moses	59.25	58.60	58.92	
Nematus	70.94	75.48	73.14	
FastText SelecT MT System				
Apertium	70.52	46.79	56.25	68.12%
Moses	60.72	60.27	60.49	
Nematus	71.86	80.30	75.84	
Logistic Regression SelecT MT System				
Apertium	71.30	37.26	48.94	65.05%
Moses	57.60	52.20	54.76	
Nematus	67.71	82.61	74.42	

Table 3: Evaluation of 3 models on 3 MT systems

Lastly, in Table 4, we report system combination scores as follows: 1) the ensemble system, **SelecT**, selects translations based on the predictive model; 2) the upper bound: always choosing the best scoring system; 3) the lower bound: always choosing the worst scoring system.

System	BLEU	WER	Unique Tokens
Best	40.08	46.70	23767
Worst	18.97	63.91	18595
RNN	37.36	49.69	24546
FastText	38.01	49.55	24790
LR	38.03	49.97	24935

Table 4: Comparison of 3 **SelecT** MT systems

Our FastText system. for example, had a 19.04 improvement over the BLEU lower-bound of (90.2% of the potential difference) and a 14.36 improvement over the WER lower-bound (83.4% of the potential difference), in both cases, this is significantly more than the average of the upper and lower bounds (29.53 BLEU score and 55.31 WER). The ensemble system (using FastText) also out-performs the best individual system (Nematus) by .65 Blue and 2.22 WER. The average between the upper and lower bounds is a good baseline to beat, to demonstrate that our system is successful at predicting the correct high-scoring system most of the time. However, being the best system gives the results practical value.

We observe that Nematus is more likely to correctly handle polysemous words (should English *march* be translated to Spanish as *marzo* (the month) or *marcha* (the action)). However, some of Nematus' errors involve seemingly arbitrary translations of words or the addition of arbitrary words. For example, the English "*identification numbers*" is correctly translated as "*números de identificación*" by Apertium, but Nematus translates it as *identificación de identificación* (Moses translates it nearly correctly, but leaves off the "s" in "*números*"). Similarly, Apartium correctly translates the English "*saffron*" as *azafrán*, whereas Moses leaves it untranslated ("*saffron*") and Nematus translates it mysteriously as "lágrimas de los perros".

6.1 FMR-based performance

We evaluate our best performing model (FastText) from 5.1.1 on the agnostic black-box MT system from FMR (Ortega et al., 2016). Table 5 shows our approach for 3 different fuzzy-match score thresholds (FMT) —60%, 70% and 80%—. For our experiments, we use a Levenshtein-based word-error rate distance measurement as described earlier. **SelecT** models are used to select translations for all potential segments (s' segments and sub-segments σ and σ' in work from Ortega et. al (2016)) when

	TM	Apertium		Moses		Nematus		SelecT	
		MT	FMR	MT	FMR	MT	FMR	MT	FMR
FMT: 60%									
Error (%)	55.0	65.3	36.5	45.8	29.2	48.6	30.1	44.8	27.9
Er. (%) on matches	20.1	65.3	17.9	45.8	16.2	48.6	17.1	44.8	16.0
# matches	1184	1993	1184	1993	1184	1993	1184	1993	1184
Avg. length	22.6	22.1	22.6	22.1	21.1	22.3	21.3	22.1	22.8
FMT: 70%									
Error (%)	61.0	65.3	38.5	45.8	30.5	48.6	31.15	44.8	29.2
Er. (%) on matches	16.3	65.3	14.6	45.8	13.7	48.6	13.9	44.8	13.5
# matches	828	1993	828	1993	828	1993	828	1993	828
Avg. length	22.4	22.1	22.5	22.1	22.8	22.2	22.8	22.1	22.7
FMT: 80%									
Error (%)	69.7	65.3	42.6	45.8	32.6	48.6	33.7	44.8	31.7
Er. (%) on matches	13.1	65.3	11.9	45.8	11.3	48.6	11.4	44.8	11.2
# matches	660	1993	660	1993	660	1993	660	1993	660
Avg. length	22.3	22.2	22.4	22.1	23.4	22.2	23.4	22.1	22.8

Table 5: Word-Error Rate (WER) evaluation for FMR using **SelecT** and black-box MT

FMR creates a hypothesis $t*$; then, FMR selects the best hypothesis according to the edit-distance between the hypothesis and the reference t'.

Like work from Ortega et. al (2016) we report on 2 error rates: 1) WER computed on the whole test set and 2) WER computed only on the segments for which a translation unit (TU) with a fuzzy-match score above a threshold is found (error on matches). We use the 2 different forms of measurement to better understand how a translator or CAT tool user would use FMR in a production setting since they would typically only see matches. It is also worthwhile to note that the scores for FMR are based on an oracle setting which implies knowledge of the reference translations (t' for each hypothesis ($t*$).

As seen in Table 5, the **SelecT** system performs better than Ortega et. al (2016). In addition to outperforming work by Ortega et. al (2016), it seems to score well when compared to other work by Knowles et. al (2018). An explanation by Knowles et. al (2018) has already been given as to why Moses performs better in certain situations. It's our belief that in addition to previous work from both authors, our prediction system scores well due to the trained knowledge it has gained from DGT-TM 2016 which is similar to DGT-TM 2015, despite the MT systems themselves being trained on Europarl V7. **SelecT** outperforms all systems in both fuzzy-match situations (matched or not). It even performs better when there's no fuzzy-match and the MT system has to translate the entire source segment (s' in Ortega et. al (2016)).

FMR (Ortega et al., 2016) has already shown to be a potential win for improving translator's productivity. The **SelecT** system presented here shows performance gains of as much as 2 points in WER over previous work (Ortega et al., 2016). We believe that the gains presented here, much like points brought up in 5.1.1, are due to Moses and Apertium's phrase-based and rule-based technology that allow it to come somewhat closer to translator's needs at the sub-segment level. Subsegments in FMR are usually shorter and have more punctuation involved (especially in the DGT-TM 2015 corpus); it's the case here that an ensemble system covers more cases than any one MT system tested and could, thus, be more valuable for a translator or CAT-tool user.

7 Conclusion

Our experiments show that **SelecT** can be used to increase performance in black-box MT tools. **SelecT** is agnostic to other processes in a typical MT pipeline and does not require underlying process changes in current black-box MT systems. **SelecT** only requires access to a command-line utility that accepts a sentence as input to select the best MT system. The work presented in section 5.1.1 also helps explain how well various models perform for black-box systems. Baseline MT systems are combined with a predictive model to create a non-traditional ensemble for improving translations from tools using black-box translation. In our experiments, FastText outperformed other models as measured by BLEU and WER. There are surely more prediction models (non-baseline) that could perform better but we leave that for future work.

8 Future Work

We are considering several avenues for future work including trying additional classifiers for choosing the best MT system including a convolutional neural network (CNN). We would also like to try additional MT systems such as OpenMT [20] or Google translate.[21] In particular, it would be nice to demonstrate whether it is as important to combine diverse systems as it is to combine high-performing systems when creating an ensemble. Our classifiers were also very similar to most baseline systems conventionally found on-line. We feel that by training the systems on more in-domain data as presented in previous work (Knowles et al., 2018), we would improve the results. The classifiers could also be trained with more information about the text very similar to the QE tasks presented by Specia et. al (2010). One could also use QE as a corner stone for leveraging systems that would not only predict via sentence-level features; but, could also predict using the other features presented at the post-editing level as done by Chatterjee et. al (2015).

9 Acknowledgements

We thank Rebecca Knowles for providing MT output using Nematus models from Knowles et al. (2018). John E. Ortega is supported by the Universitat d'Alacant and the Spanish government through the EFFORTUNE (TIN2015-69632-R) project. Kyunghyun Cho was partly supported by Samsung Advanced Institute of Technology (Next Generation Deep Learning: from pattern recognition to AI) and Samsung Electronics (Improving Deep Learning using Latent Structure).

References

Alva-Manchego, F., J. Bingel, G. Paetzold, C. Scarton, and L. Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 295–305.

Arenas, A. G. 2013. What do professional translators think about post-editing. *The Journal of Specialized Translation*, (19).

Bentivogli, L., A. Bisazza, M. Cettolo, and M. Federico. 2016. Neural versus phrase-based machine translation quality: a case study. *CoRR*, abs/1608.04631.

Bojar, O., R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, Antonio Jimeno Y., P. Koehn, V. Logacheva, C. Monz, M. Negri, A. Neveol, M. Neves, M. Popel, M. Post, R. Rubino, C. Scarton, L. Specia, M. Turchi, K. Verspoor, and M. Zampieri. 2016. Findings of the 2016 conference on machine translation. In *WMT16*, pages 131–198.

Bojar, O., R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, et al. 2017. Findings of the 2017 conference on machine translation (wmt17). In *WMT17*, pages 169–214.

Chatterjee, Rajen, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the planet of the apes: a comparative study of state-of-the-art methods for mt automatic post-editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 156–161.

Chatterjee, R., J. GC de Souza, M. Negri, and M. Turchi. 2016. The fbk participation in the wmt 2016 automatic post-editing shared task. In *WMT16*, pages 745–750.

Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Cutting, D., J. Kupiec, J. Pedersen, and P. Sibun. 1992. A practical part-of-speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 133–140.

Dugast, L., J. Senellart, and P. Koehn. 2007. Statistical post-editing on systran's rule-based translation system. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 220–223.

Forcada, M. L, B. I. Bonev, S Ortiz Rojas, J. P. Ortiz, G R. Sánchez, F S. Martínez, C. Armentano-Oller, M. A Montava, and F. M Tyers. 2009. Documentation of the open-source shallow-transfer machine translation platform apertium. *Online Departament de Llenguatges i Sistemes Informatics Universitat d Alacant, Available: http://xixona. dlsi. ua. es/˜ fran/apertium2-documentation.*

Forcada, M. L., Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M. Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144.

Frederking, R. and S. Nirenburg. 1994. Three heads are better than one. In *Proceedings of the ANLP*.

[20] http://opennmt.net/

[21] http://translate.google.com

Heafield, K. and A. Lavie. 2010. Combining machine translation output with open source: The Carnegie Mellon multi-engine machine translation scheme. *The Prague Bulletin of Mathematical Linguistics*, 93:27–36.

Junczys-Dowmunt, M., T. Dwojak, and H. Hoang. 2016. Is neural machine translation ready for deployment? a case study on 30 translation directions. *arXiv preprint arXiv:1610.01108*.

Knowles, R., J. E Ortega, and P. Koehn. 2018. A comparison of machine translation paradigms for use in black-box fuzzy-match repair. In *AMTA 2018*, volume 1, pages 249–255.

Koehn, P. and R. Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Koehn, P., 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. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180.

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

Lavie, A. and A. Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *StatMT '07*, pages 228–231.

Luong, T., I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba. 2014. Addressing the rare word problem in neural machine translation. *CoRR*, abs/1410.8206.

Nair, V. and G. E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML-10*, pages 807–814.

Nomoto, T. 2004. Multi-engine machine translation with voted language model. In *ACL '04*.

Ortega, J. E, F. Sánchez-Martınez, and M. L Forcada. 2014. Using any machine translation source for fuzzy-match repair in a computer-aided translation setting. In *AMTA 2014*, volume 1, pages 42–53.

Ortega, J. E., F. Sánchez-Martínez, and M. L. Forcada. 2016. Fuzzy-match repair using black-box machine translation systems: what can be expected? In *AMTA 2016, vol. 1*, pages 27–39.

Rosti, Antti-Veikko, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr. 2007. Combining outputs from multiple machine translation systems. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 228–235.

Salton, G. and C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513 – 523.

Sánchez-Cartagena, V. M. and A. Toral. 2016. Abumatran at wmt 2016 translation task: Deep learning, morphological segmentation and tuning on character sequences. In *WMT16*.

Schwenk, H., A. Rousseau, and M. Attik. 2012. Large, pruned or continuous space language models on a gpu for statistical machine translation. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pages 11–19.

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

Sennrich, R., O. Firat, K. Cho, Alexandra Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. L"aubli, A. V. Miceli Barone, J. Mokry, and M. Nadejde. 2017. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Demonstrations at the 15th Conference of the EACL*.

Specia, L., D. Raj, and M. Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine Translation*, 24(1):39–50.

Specia, L., K. Shah, J. G.C. de Souza, and Trevor Cohn. 2013. QuEst - a translation quality estimation framework. In *51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84, Sofia, Bulgaria.

Wagner, R. A and M. J Fischer. 1974. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.

White, J. S. 1995. Approaches to black box mt evaluation. In *Proceedings of Machine Translation Summit V*, volume 10.