Romanian to English Automatic MT Experiments at IWSLT12
(System Description Paper)

Ştefan Daniel Dumitrescu, Radu Ion, Dan Ştefănescu, Tiberiu Boroş, Dan Tufiş

Research Institute for Artificial Intelligence
Romanian Academy, Romania
{sdumitrescu,radu,danstef,tibi,tufis}@racai.ro

Abstract

The paper presents the system developed by RACAI for the ISWLT 2012 competition, TED task, MT track, Romanian to English translation. We describe the starting baseline phrase-based SMT system, the experiments conducted to adapt the language and translation models and our post-translation cascading system designed to improve the translation without external resources. We further present our attempts at creating a better controlled decoder than the open-source Moses system offers.

1. Introduction

This article presents the system developed by RACAI (the Research Institute for Artificial Intelligence of the Romanian Academy) for the ISWLT 2012 competition. We targeted the Machine Translation track of the TED task, Romanian to English translation.

We had access to the following resources:

- In-domain parallel corpus: 142K sentences; 13MB size; TED RO-EN sentences [6].
- Out-of-domain parallel corpus: 550K sentences; 85MB size; Europarl (juridical domain) and SETimes (news domain) RO-EN sentences.
- Out-of-domain monolingual corpus (English): 168M sentences; 26GB size; mostly news domain EN sentences.
- Development set: 1.2K RO-EN sentences (TED tst2010 file)
- Test set: 3K RO only sentences (TED tst2011 and tst2012 files).

Before attempting any translation experiments, the available resources had to be preprocessed. This involves first correcting the Romanian side of the parallel corpora as to obtain the highest possible quality Romanian-side text and then annotate both the Romanian and English sides.

Thus, the first preprocessing step involves automatic text normalization. Historically, due mainly to technical reasons regarding the code-page available in earlier versions of the Windows operating system, the letters *ș* and *ț* in the Romanian language were initially written as **ş**, **ţ** (with a cedilla underneath – old, incorrect style) and later as **ș**, **ț** (with a comma underneath – correct style). As such, we have several resources with incompatible diacritics for these two letters. All old-style letters have been converted to the new style. The second correction to be made is due to the Romanian orthographic reform from 1993 which re-establish the orthography used until 1953, according to which (among the others) the inner letter “î”, has been replaced by “â (ex: p**î**ine is written correctly as p**â**ine). Older texts have been corrected to the current orthography using an internally developed tool that uses a 1.5 million word lexicon of the Romanian language backing-off a rule-based word corrector in case the lexicon might not contain some words.

The third and final necessary correction concerned texts that do not have diacritics. In the provided resources, both in-domain and out-of-domain corpora contain several groups of sentences that have not diacritics. Restoring diacritics is a rather difficult task, as a misplaced or missing diacritic can have dramatic effects starting from change of definiteness of a noun (for example) to changing an entire part-of-speech of a word, yielding sentences that lose their meaning. Using an internally developed tool [19] we were able to carefully restore diacritics where they were missing. Even though the tool is not 100% accurate, it is better to introduce a small amount of error rather than have several words without diacritics that will create more uncertainty in the translation process later on.

The second step of the preprocessing phase is the automatic annotation of both Romanian and English texts. Using also an internally developed tool named TTL [11] we are able to tokenize sentences and annotate each word with its lemma, two types of part-of-speech tags: morpho-syntactic descriptors (MSDs) and a reduced tag set (CTAGs), and different combinations of them. The tags themselves follow the Multext-East lexical standard [8] and the tiered tagging design methodology [20].

As an example, for the English sentence “We can can a can.” we obtain the following annotation:

We|we^Pp|we^PPER1|Pp1-pn|PPER1
can|can^Vo|can^VMOD|Voip|VMOD
can|can^Vm|can^VINF|Vmn|VINF
a|a^Ti|a^TS|Ti-s|TS
can|can^Nc|can^NN|Ncns|NN
.|. ^PE|.^PERIOD|PERIOD|PERIOD

The first of the five factors for each word is the word itself (the surface form). The second factor is the lemma of the word, linked by the “^” character, to its first two positions in the MSD tag (grammar category and type). The third factor is the lemma linked to the CTAG, followed by the MSD (fourth factor) and CTAG (fifth factor).

The TTL tool has other advanced features that make it desirable for machine translation. Sometimes it is better for certain phrases to be considered as a single entity. For example, phrases like "… do something to **the other**, …" are automatically linked together by an underscore and annotated as: "the_other|the_other^Pd|the_other^DMS|Pd3-s|DMS". Other examples of automatically extracted phrases: "in_terms_of", "the_same", "a_little", "a_number_of", |"out_of", "so_as", "amount_of_money", "put_down", "dining_room", etc. The same tokenization, phrase extraction and annotation process is performed for the Romanian language.

The third and last step of the preprocessing phase is true-casing all available resources. True-casing simply means lower-casing the first word in every sentence, where necessary. A model is trained on available data, learning what words should not be lower-cased, as acronyms or proper nouns, and applied back to the data. True-casing benefits automatic machine translation when building both the translation model and the language model by reducing the number of surface forms for each possible word.

2. System description

In this section we present the steps and the experiments performed to create and adapt our MT system to the TED task. We start with a basic phrase-based statistical MT system with default parameters in order to establish a baseline (section 2.1); we then experiment with different adaptations of the language models and the translation tables used (2.2 – 2.4); we perform a parameter setting search to find the combination of parameters that will maximize the translation score (2.5); finally, we apply a technique we call "cascaded translation" [21] to attempt to correct some of the translation errors (section 2.6).

Before describing the steps and experiments performed, we must specify that unless explicitly otherwise stated, the following BLEU scores are all obtained on comparing the English translation of the tst2012 file from the test set to an English reference file we manually created starting from the English subtitles for each respective TED talk. We later obtained access to the English tst2011 file from the same test set, but we did not have enough time to re-run the experiments on this official reference file. We are confident that our tst2012 reference file is very similar to the official file given the correlated scores of our results and those given by the official evaluation as we later present.

2.1. Baseline system

We start with the standard Moses [12] system. We trained the system on the in-domain data (the provided TED RO-EN parallel corpus), as well as building a language model on the English side of the same corpus.

The language model was built using the SRILM toolkit [17]: surface-form, 5-gram, interpolated, using Knesser-Ney's smoothing.

This baseline system yielded a 25.34 BLEU score.

2.2. Direct Language-Model adaptation experiment

The first attempted language model adaptation method is the direct, perplexity-based measure: given the tokenized and true-cased English resources, extract sentences with the lowest perplexity and add them to the in-domain language model.

The procedure first requires that all the English resources (both from the parallel corpora and the monolingual corpora) be merged into a single file. The resulting 27 GB file had around 28 billion tokens contained in almost 168 million sentences. Each sentence was perplexity measured against the in-domain language model. Then, the file was sorted based on sentence perplexity, lowest first.

Starting with the initial in-domain language model that obtained 25.34 BLEU points we added incrementally batches of 1 million sentences, re-translated and noted the score increase/decrease. We observed a non-linear increase up to 10 million added sentences, followed by a rather slow BLEU decrease. We found that the best performing language model constructing using this method contains 10.6 million sentences, 142,000 coming from English side of the in-domain corpus. The score obtained using this method was 28.04, a significant 2.70 point increase from the baseline score of 25.34.

2.3. Indirect Language-Model adaptation experiment

The direct language model adaptation works very well when a specific domain is given and a language model can be built on that domain to provide a perplexity reference for new sentences. If this information is not available, one could try to alleviate the problem in various ways.

Our idea in this indirect language model adaptation is to check whether we could use the information available in the test set to create a better language model.

This, however, presented a problem: while in the test set we are only given the source Romanian sentences that need to be translated, the English language model should be adapted with sentences for which translations are not yet available. Thus, we came up with the following four step procedure to attempt indirect adaptation of the target language model by generating English n-grams from Romanian n-grams:

Step 1: Count the n-grams from the Romanian sentences in the test set. Counting was done up to 5-grams, ignoring functional unigrams (determiners, prepositions, conjunctions, etc.).

Step 2: Having the translation table already created from the base model, attempt to "translate" the n-grams from Romanian to English. Parse the translation table, look up each Romanian n-gram and retain all the equivalents in English. This will increase the number of n-grams several times. At the end of this step we will have a list of English n-grams.

Step 3: Based on the list of English n-grams, iterate over each sentence in the file containing all the English data (27 GB) and count matching n-grams. In order to select the most promising sentences, we have created a few different scoring methods: **(1)** Standard measure, where if we find a matching n-gram we increase the score of that sentence by n (e.g. if we find four unigrams and two trigrams we increase the score by 4*1+2*3 = 10); **(2)** Standard normalized (Std. Div.) measure, where we divide the standard measure by the length of the sentence in order to compensate for very long sentences likely to have more n-gram matches; **(3)** Square measure, where if we find a matching n-gram we increase the score of the sentence by the square of n (ex: for 4 unigrams and two trigram the score would be $4*1^2+2*3^2=22$); **(4)** Square normalized (Square Div.) measure, dividing the Square measure by the length of the sentence in order to compensate for long sentences. We thus sort in decreasing order each of the English sentences based on our proposed measures, obtaining 4 large English files.

Step 4: From each of the four sorted files, we take incremental batches of sentences and build adapted language models of larger and larger sizes.

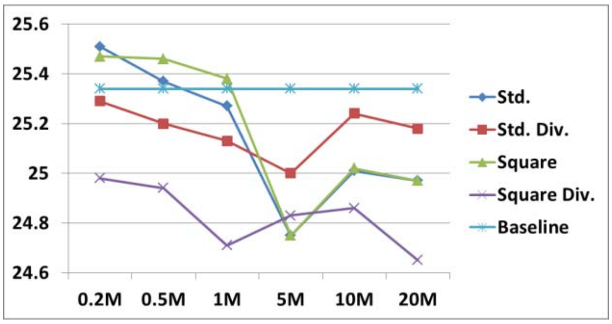

Figure 1: Indirect LM adaptation BLEU scores

Figure 1 presents our experimental results. We manage to obtain just a very slight increase over the baseline of 25.34 when adding just a small number (less than 200,000 sentences in addition to the TED English sentences). This experiment shows that it is possible to adapt a language model starting only from the sentences that need to be translated, but also reveals that there is a fine-grained point over which adding more sentences, using our measures, actually degrades performance. Also, it should be noted that for both direct adaptation using the perplexity measure and the indirect adaptation method, the peak of the graph can be determined only if the target (reference) development set, on which to measure the BLEU score, is available. However, our indirect LM adaptation allows increasing the size of the available development set considering the monolingual test set.

2.4. Translation model adaptation experiment

With the next experiment we attempt to adapt the translation model (TM) using data available from the out-of-domain corpora.

Based on the previous experiments we used perplexity as the similarity measure of choice. We attempted two adaptations based on both the source and the target languages. We built two language models: the first was built on the English side of the TED corpus while the second on the Romanian side. Using each language model in turn, we calculated the perplexity of each corresponding sentence from every translation unit in the out-of-domain parallel corpora. Then we sorted the corpora's translation units according to the perplexity scores of English and Romanian parts. For example, we measured the perplexity of the Romanian side of Europarl & SETimes corpora vs. the language model built on the Romanian side of TED, and then sorted Europarl & SETimes by the ascending perplexity of their Romanian sides (similarly for English).

We made experiments on TM adaptation selecting parallel data according to the similarity with each language model. We took increments of 5% of the sorted parallel corpora and added them to the TED corpus and noted the translation scores. For this experiment we used the development set (tst2010) which had a translation baseline score of 28.82.

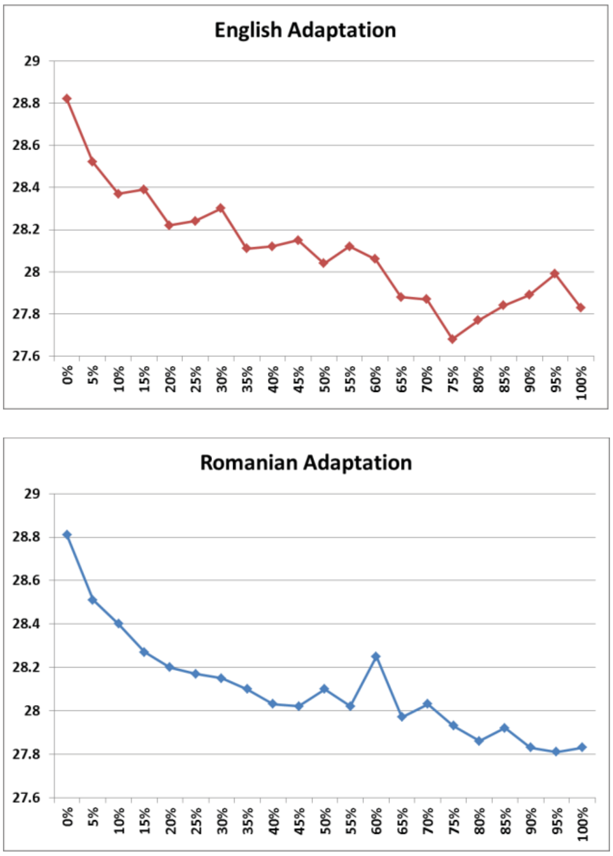

Figure 2: English and Romanian TM adaptation graphs

The experiments show that even adding 5% of the best sentences (based on perplexity) of the Europarl and SETimes corpora decreases the translation score by a significant 0.3 BLEU points. The decrease is rather consistent when trying to adapt the translation model starting from either the Romanian or the English language, clearly stating the conclusion that neither Europarl which is a juridical corpus nor SETimes which is news-oriented do contain parallel sentences that positively contribute to the translation model firmly located in a free-speech domain. After this result it was clear that further attempting to adapt the translation model using the provided out-of-domain corpora was impractical. Using the LEXACC comparable data extraction tool [18] with the TED and Europarl+Setimes corpora as search space supported the previous observation that the out-of-domain data was too distant from the in-domain-data to be useful in TM adaptation.

2.5. Finding the best translation system

Having experimented with adapting both the language model and the translation model, we started searching for the parameter combination that will maximize the translation score.

The systematic search included the following parameters:

- Translation type
- Alignment model
- Reordering model
- Decoding type and sub-parameters

The translation type refers to which word factors were used and the translation path itself. We started from the simple surface-to-surface translation, gradually using more factors such as part-of-speech (both MSDs and CTAGs, available after using the TTL tool in the corpus preprocessing phase), lemma or different combinations of lemmas and part-of-speech tags. The translation path meant using direct, single-step translation (ex: translation of surface-surface, translation of surface and part-of-speech to surface, etc.) or multiple step translation including generation phases (ex: translation of lemma to lemma then generation of part-of-speech from lemma, then translation of part-of-speech to part-of-speech and finally generation of the surface form from lemma and part-of-speech).

For the alignment and reordering models we also tried using several combinations of word factors.

Finally, for the decoder, we systematically modified the decoding parameters for the default decoder (beam size, stack size) and the decoding model (cube-pruning, minimum-bayes-risk and lattice-minimum-bayes-risk, each with its individual parameters).

After conducting an extended search of about 60 experiments in which parameters were systematically modified we obtained a score of 29.24, again a significant increase from the baseline system with the adapted language model for which we obtained only 28.04. These two figures are unofficial results computed (as mentioned in Section 2) on our hand made reference for tst2012. The best combination of parameters was: a single-step direct translation of surface form to surface form; an alignment model using the "*union*" heuristic; a reordering model using the default "*wbe-msd-bidirectional-fe*" heuristic; the alignment and reordering model based only on the lemma and the reduced MSD, not on the surface forms; a lattice-minimum-bayes-risk decoder with an increased stack size of 1000.

The search was performed using the adapted language model described in section 2.2 and a translation model based only on the TED in-domain corpus.

2.6. Cascaded system translation experiment

Having obtained the optimum parameters so far, we applied a procedure we previously developed [21] to try to further improve the translation score without adding or using any external data. We hypothesize that training a second phrase-based statistical MT system on the data that was output by our initial system, this second system will correct some of the errors the initial system made.

The first step in building the second system of the cascade is based on using the first system to translate the Romanian side of its own RO-EN training corpus. This will yield a translated–EN-EN parallel corpus on which the second system is trained upon. The cascaded system is now ready to be used.

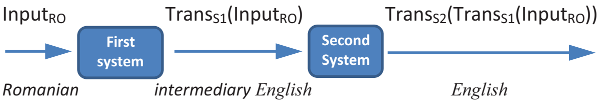

Figure 3: Cascaded system diagram

The diagram shows how the cascading procedure works. The test set is initially translated from Romanian into intermediary English. Next, this intermediary translation is fed to the second system which translates the intermediary English to "final" English. The "final" English is then evaluated against the reference to determine the effect of the cascade: how much improvement was achieved, if any.

We obtained a net increase of 0.36 points bringing the new BLEU score to 29.60 (using our tst2012 manually created reference file). In this particular case the cascade changed 22 percent of the total of 1733 sentences, 12% for the better and 10% for the worse, the rest of the sentences being unaffected.

Table 1: Cascading effect

S1	After system 1	S2	After system 2	Reference
0.57	the microprocessor **.** **it 's a miracle** the personal computer is a miracle .	1.00	the microprocessor **is a miracle .** the personal computer is a miracle .	the microprocessor is a miracle . the personal computer is a miracle .
0.53	and the reasons **delincvenților** online are very easy to understand .	0.7	and the reasons **online criminals** are very easy to understand .	and the motives of online criminals are very easy to understand .
0.47	and **so let me** begin with an example .	0.31	and **let me try to** begin with an example .	and let me begin with one example .

Table 1 shows some of the effects of cascading. In the first example we see a clear improvement from 0.57 to 1.00 of the translation by correctly placing the comma and transforming "it's a" in "is a". The second example shows that sometimes the cascade can correct initially non-translated words: due to Moses's phrase table pruning mechanism, even though the unigram "delincvenților" is present in the training corpus, it does not appear in the first system's phrase table and thus does not get translated. However, it appears in the second phrase table and is subsequently translated. The third example presents a score decrease from 0.47 to 0.31. However, transforming "so let me" to "let me try to", while from

BLEU's perspective vs. the reference translation is a decrease, from a human perspective, the sentence is still fully comprehensible.

Overall, cascading increases the BLEU score usually from a fraction of a BLEU point up to a few BLEU points [21]. For the official evaluation we have submitted for each test file a cascaded system and a non-cascaded system. The official evaluations showed a small increase of 0.04 BLEU (from 29.92 for the standard, un-cascaded system to 29.96 for the cascaded) for the 2011 test file and an increase of 0.21 BLEU (from 26.81 to 27.02) for the 2012 test file, as presented in Table 2 in Section 4.

3. Alternative translation systems

After performing a host of experiments with Moses with different settings as reported in the previous sections, it became clear that the BLEU barrier of around 30% is not going to be easily (and significantly) broken without additional in-domain, parallel data and because of that, we proceeded to refine our own, in-house developed decoders based on Moses-trained phrase tables and language models. The purpose of this endeavor was to come up with a combination/merging scheme of the outputs of several decoders that, we envisaged, would ensure a superior translation when compared to each of the decoders. In what follows, we briefly give the underlying principles of our in-house developed decoders and present their combined output with the best Moses output (see 2.6).

3.1. The first RACAI decoder (RACAI1)

The first RACAI decoder is based on the Dictionary Lookup or Probability Smoothing (DLOPS) algorithm [4], primarily used for phonetic transcription of out-of-vocabulary (OOV) words. The original algorithm works by adjoining adjacent overlapping sequences of letters that have corresponding transcription equivalents inside a lookup table. The overlapping sequences are selected by finding a single split position (called *pivot*) inside a sequence that will maximize a function called the *fusion score* (described in the original article). The algorithm would recursively produce the phonetic transcriptions of the pivot left and right sequences either by directly returning transcription candidates from the lookup table (if there are any transcription candidates) or by further recursive building the transcriptions. Because of the similarities that arise between the phonetic transcription and MT [13], we thought of adapting DLOPS to perform decoding for MT. There were some limitations of the initial algorithm that needed to be eliminated:

1. We modified the system to use a Berkeley Data Base (BDB) for lookup to be able to cope with large phrase tables;
2. The algorithm looks for the sequence of words with the highest translation score. The indexes of the left-most and right-most words are considered the pivots of the recursions. The DLOPS had to be modified to search for two pivots instead of one;
3. We added word reordering capabilities (this was not an issue in phonetic transcription).

For each sequence of words that has a corresponding entry in the translation table, we retain all possible candidates and, returning from the recursive call, we get the Cartesian product of the translations from the left, center and right source word sequences. Because this translation set usually has a large number of candidates, we score each translation candidate by summing the *S* value for the left, center and the right sub-candidate:

$$S = \theta_1\varphi(f \mid e) + \theta_2\varphi(e \mid f) + \theta_3\lambda(f \mid e) + \theta_4\lambda(e \mid f) + \theta_5 LM(e)$$

where $\varphi(f \mid e)$ is the Moses-based phrase table inverse phrase translation probability, $\varphi(e \mid f)$ is the direct phrase translation probability, $\lambda(f \mid e)$ is the inverse lexical similarity score, $\lambda(e \mid f)$ is the direct lexical similarity score and $LM(e)$ is the language model score (at word level) of the translation candidate. The weights $\theta_{1,\ldots,5}$ are computed with the Minimum Error Rate Training (MERT) procedure from the Z-MERT package [23].

3.2. The second RACAI decoder (RACAI2)

This first step of this decoder is to collect a set *C* of source sentence non-overlapping segmentations according to the phrase table, giving priority to segmentations formed with the longer spans of adjacent tokens from the input sentence. For the input sentence *S* with *n* tokens, considering at most *k* adjacent tokens (called "a token span") for which we find at least one translation in the phrase table, *k* < *n*, the total number *N* of non-overlapping segmentations is

$$N_k(n) = \sum_{i=1}^{k} N_k(n-i)$$

For k = 2 this is the well-known Fibonacci series and it is obvious that $N_k(n) > N_2(n)$ for *k* > 2. It can be shown that

$$N_2(n) \geq c\left(\frac{3}{2}\right)^n$$

for some positive constant *c* and this tells us that one cannot simply enumerate all the segmentations of the source sentence according to the phrase table because the space is exponentially large. Thus, our strategy is to choose a segmentation $P = \left\{w_i^j \mid 1 \leq i < j \leq n\right\}$, where w_i^j is the token span from the index *i* to index *j* in the source sentence *S* which has at least one translation in the phrase table, such that $|P|$ is minimum.

The second step of the decoder is to choose, for each partial translation h_1^j (up to the current position *j* in *S*) and input token span $w_{j+1}^k \in P$, the best translation h_{j+1}^k from the phrase table such that two criteria are simultaneous optimized:

1. The translation scores of h_{j+1}^k from the Moses phrase table are maximum;
2. The language model (at word form level and POS tag level) score of joining h_1^j with h_{j+1}^k is also maximum.

What we did, was to actually compute an interpolated score as in the case of the previously described decoder with weights tuned with Z-MERT.

The third and final step of the RACAI2 decoder was to correct the raw, statistical translation output to eliminate the translation errors that were observed to be frequent and that violate the English syntactic requirements (mainly due to the inexistence of a reordering mechanism). This is a rule-based module that works only for English. Examples of frequent mistakes include:

- translating the valid sequence "noun, adjective" from Romanian into the same, invalid, sequence in English;
- translating the valid sequence "noun, demonstrative determiner" from Romanian into the same, invalid, sequence in English;
- translating the valid sequence "noun, possessive determiner" from Romanian into the same, invalid, sequence in English.

The astute reader has noticed that the optimization criteria from the second step of this decoder consider local maxima. One immediate improvement is to replace the current optimization step by a Viterbi global optimization [22].

3.3. Combining translations from Moses, RACAI1 and RACAI2

Having three decoders that produce different translations for the same text, it is tempting to consider their combination in order to find a better translation. Generating the best translation for a text (sentence or paragraph), given multiple translation candidates obtained by different translation systems, is an established task in itself. Even the simplest approach of deciding which candidate is the most probable translation has been proven to be difficult [1, 5, 16]. The different solutions described in the literature are focused on re-ranking merged N-best lists of translation candidates, word-level and phrase-level combination methods [2, 6, 8, 14].

Our approach is a phrase-level combination method and exploits the linearity of the candidate translations given by the systems we employed. First, we split the source (i.e. Romanian) sentence into smaller fragments which are considered to be stand-alone expressions that can be translated without additional information from the surrounding context. For considerations regarding speed, this is done by using certain punctuation marks and a list of words (split-markers) that can be considered as fragment boundaries (e.g. certain conjunctions, prepositions, etc.). Every fragment must contain at least two words, out of which one should not be in the above mentioned list of split-markers. For example, the sentence "*s-a făcut de curând un studiu printre directorii executivi în care au fost urmăriţi timp de o săptămână.*"[1] is split into 3 fragments: "*s-a făcut de curând un studiu*", "*printre directorii executivi*" and "*în care au fost urmăriţi timp de o săptămână.*"

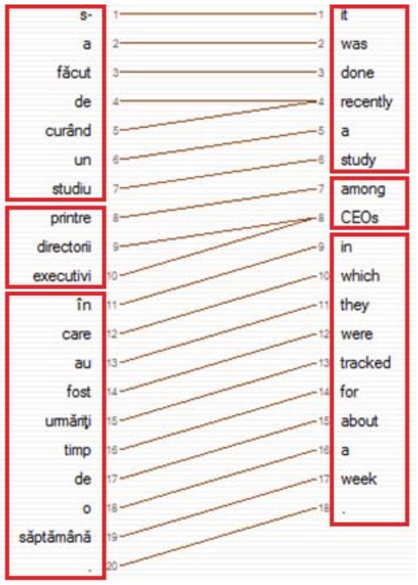

Figure 4: DTW Alignment helps identifying the corresponding translations of the source fragments

In the next step, taking into account the linearity of the translations, we use Dynamic Time Warping (DTW) algorithm [3,15] to align the source sentence with the current translation candidate. The cost function is defined between a source word w_s and a target word w_t as: $c = 1 - te(w_s, w_t)$, where te is the translation equivalence score in the existing dictionary. Taking into account the source fragments and the alignments obtained with DTW, we are able to pinpoint the translation for each of fragment. For our example we have the following candidates:

Table 2: Translation candidates for the source fragments

Translation/ system	*s-a făcut de curând un studiu*	*printre directorii executivi*	*în care au fost urmăriţi timp de o săptămână.*
Moses	it has recently made a study	**among the CEOs**	in which they were followed for about a week.
RACAI 1	**it was done recently a study**	among CEOs	in which they were tracked for about a week.
RACAI 2	was done recently a study	among execs executives	**in which have been tracked for about a week.**

We modeled the selection process by a HMM. The emission probabilities are given by a translation model learned with Moses, while the transition probabilities are given by a language model learned using SRILM. The combiner uses the Viterbi algorithm [22] to select the best combination of the translation candidates and generate a "better" translation. For our example, the best path found by the Viterbi algorithm passes through the bolded fragments in the above table, yielding the final translation: "it was done recently a study among the CEOs in which have been tracked for about a week.". Yet, this translation is deficient because of the missing

[1] English: "there was also a study done recently with CEOs in which they followed CEOs around for a whole week."

pronoun “they” (existing in Moses and RACAI1 outputs) in the translation for the third fragment.

We have also experimented with combination at the whole-translation (sentence) level (as opposed to phrase-level) and we tried the following:

1. selecting the translation which had the lowest perplexity as measured by the language model of the best Moses setting;
2. selecting the translation which had the largest averaged BLUE score when compared to the other two translations;
3. selecting the translation which had the lowest TERp score when compared to its cascaded version.

The phrase-level combination method outperforms the first sentence-level combination method and it is close (somewhat better) to the other two sentence-level combination methods. We also estimated the maximum gain (an “oracle” selection) from the sentence-level combination by choosing the translation which had the highest BLUE against our reference for tst2012 (see Table 3). We have thus determined the 32.41 BLUE score which is 2.81 points better than the cascaded Moses (29.60).

Even if the phrase-level combination method does not outperform Moses, our analysis shows that the combiner improves about 22% of the Moses translations with an average increase of the BLEU score of 0.088 points per translation while it deteriorates about 27% of them with an average decrease of the BLEU score of 0.098 points per translation, amounting to a global decrease of only 0.69 BLEU points overall (see Table 3; compare S2 with S5). The rest of the translations remained unchanged after the combination.

4. Conclusions

The paper presented RACAI’s machine translation experiments for the IWSLT12 TED track, MT task, Romanian to English translation. In the first part we presented our experiments in building a system based on the Moses SMT package. We evaluated different adaptation types for the language and translation model; we then performed a systematic search to determine the best translation parameters (word factors used, alignment and reordering models, decoder type and parameters, etc.); finally, we applied our cascading model to correct some translation errors made by our best single-step translator. This experiment chain yielded our best model, in the official evaluation (Table 2) obtaining 29.96 BLEU points for the tst2011 test set and 27.02 BLEU point for the tst2012.

The second part of the paper presents our experiments in building two prototype decoders and a translation combiner. The decoders (RACAI 1&2) are based on different strategies than Moses (each presented in its own section), in our attempt to go beyond the difficult to reach baseline set by the best Moses-based model. However, even though we could not exceed yet this baseline, we came rather close to it, given that most of the development work was on adapting the Moses model and allowing only around 3 weeks for the development of the alternative decoders.

The following tables show the official results [9] (case and punctuation included) for the entire test set (tst2011&2012), as well as the results obtained on the reference we built for tst2012 (the official reference was not released at the time of this writing). The tables contain the performance figures for our two Moses-based models (S1 being the best direct translation model we found, while S2 being the S1 model with our cascading technique applied), our two prototype decoders (S3 and S4) and our translation combiner (S5).

Because we have not seen the reference for tst2012, our explanation for the differences among the figures in Table 2 and Table 3 is that our evaluations were performed on lower-case version of the data and mainly due to a different tokenization. While the official tokenization is based on space separation, our tokenization is language aware, considering (among others) multiword expressions and splitting clitics.

Table 2: Official systems evaluation results (case+punctuation)

System	tst2011			tst2012		
	BLEU	Meteor	TER	BLEU	Meteor	TER
S1 (Moses, not-cascaded)	29.92	0.6856	46.388	26.81	0.6443	50.891
S2 (Moses, cascaded)	**29.96**	**0.6844**	**46.701**	**27.02**	**0.6446**	**51.093**
S3 RACAI1	25.31	0.6484	48.845	22.56	0.6085	52.964
S4 RACAI2	-	-	-	21.69	0.6009	56.950
S5 Moses + RACAI1 + RACAI2	-	-	-	25.99	0.6378	51.580

Table 3: Local systems evaluation results (language aware tokenization+no case+punctuation)

System	tst2012
	BLEU
S1 = Moses, not-cascaded	29.24
S2 = Moses, cascaded	29.60
S3 = RACAI1	24.50
S4=RACAI2	23.89
S5 = Moses + RACAI1 + RACAI2	28.91
S6 = Oracle Moses + RACAI1 + RACAI2	**32.41**

5. Acknowledgements

The work reported here was funded by the project METANET4U by the European Comission under the Grant Agreement No 270893.

6. References

[1] Akiba, Yasuhrio, Taro Watanabe, and Eiichiro Sumita. 2002. Using Language and Translation Models to Select the Best among Outputs from Multiple MT systems. In Proc. of Coling, pp. 8–14.

[2] Antti-Veikko I. Rosti, Bing Xiang, Spyros Matsoukas, Richard Schwartz, Necip Fazil Ayan, and Bonnie J. Dorr. 2007. Combining outputs from multiple machine translation systems. In Proc. NAACL-HLT 2007, pp. 228–235.

[3] Bellman R. and Kalaba R. 1959. On adaptive control processes, Automatic Control, IRE Transactions on, vol. 4, no. 2, pp. 1-9.

[4] Boroş T., Ştefănescu, D., Ion, R., 2012. Bermuda, a data-driven tool for phonetic transcription of words, in Proceedings of the Natural Language Processing for Improving Textual Accessibility Workshop (NLP4ITA), LREC2012, Istanbul, Turkey, 2012

[5] Callison-Burch, Chris and Raymond S. Flournoy. 2001. A Program for Automatically Selecting the Best Output from Multiple Machine Translation Engines. In Proc. MT Summit, pp. 63–66.

[6] Cettolo, M., Girardi, C., Federico, M., *WIT3: Web Inventory of Transcribed and Translated Talks*. In Proc. of EAMT, pp. 261-268, Trento, Italy, 2012

[7] Matusov E., Ueffing N., and Ney H., 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment, in Proc. EACL, 2006.

[8] Erjavec, T., Monachini, M. 1997. *Specifications and Notation for Lexicon Encoding.* Deliverable D1.1 F. Multext-East Project COP-106. http://nl.ijs.si/ME/CD/docs/mte-d11f/.

[9] Federico, M., Cettolo, M., Bentivogli, L., Paul, M., Stuker, S.,: *Overview of the IWSLT 2012 Evaluation Campaign*, In Proc. of IWSLT, Hong Kong, HK, 2012

[10] Frederking R., Nirenburg S. 1994. Three heads are better than one. In *Proc. ANLP*, pages 95–100.

[11] Ion, R. 2007. *Word Sense Disambiguation Methods Applied to English and Romanian,* PhD thesis (in Romanian). Romanian Academy, Bucharest, 2007.

[12] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E., Moses: Open Source Toolkit for Statistical Machine Translation, in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, demonstration session, Prague, 2007

[13] Laurent Antoine, Deléglise Paul and Meignie, Sylvain. 2009. Grapheme to phoneme conversion using an SMT system. In Proceedings of INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association, pp. 708--711, Brighton, UK.

[14] S. Bangalore, G. Bordel, & G. Riccardi. 2001. Computing consensus translation from multiple machine translation systems, in Proc. ASRU, 2001.

[15] Senin P. 2008. Dynamic time warping algorithm review, University of Hawaii at Manoa, Tech. Rep.

[16] Zwarts S., Dras M., 2008. Choosing the Right Translation: A Syntactically Informed Classification Approach. In Proc. of Coling, pp. 1153-1160.

[17] Stolcke*,* A., SRILM - An Extensible Language Modeling Toolkit, in *Proc. Intl. Conf. Spoken Language Processing*, Denver, USA, 2002.

[18] Ştefănescu, D., Ion, R., and Hunsicker, S. 2012. *Hybrid Parallel Sentence Mining from Comparable Corpora*. In Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT 2012), pp. 137—144, Trento, Italy, May 28-30, 2012

[19] Tufiş, D. and Ceauşu, A., DIAC+: A Professional Diacritics Recovering System, in *Proceedings of LREC 2008*, May 26 - June 1, Marrakech, Morocco. ELRA - European Language Resources Association, 2008.

[20] Tufiş, D., Tiered Tagging and Combined Classifiers, in F. Jelinek, E. Nöth (eds) Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence 1692, Springer, 1999, pp. 28-33

[21] Tufiş, D. and Dumitrescu, S.D., Cascaded Phrase-Based Statistical Machine Translation Systems, *in Proceedings of the 16th Conference of the European Association for Machine Translation*, Trento, Italy, 2012.

[22] Viterbi, A.J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13 (2): 260–269. doi:10.1109/TIT.1967.1054010. (note: the Viterbi decoding algorithm is described in section IV.)

[23] Zaidan, O.F., 2009. *Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems*. The Prague Bulletin of Mathematical Linguistics, No. 91:79–88.