From the Real World to Real *Words*: The METEO case

Philippe Langlais, Thomas Leplus
Simona Gandrabur and Guy Lapalme

RALI
Université de Montréal
http://rali.iro.umontreal.ca

Abstract. Machine Translation (MT) is the focus of extensive scientific investigations driven by regular evaluation campaigns, but which are mostly oriented towards a somewhat artificial task: translating news articles into English. In this paper, we investigate how well current MT approaches deal with a real-world task. We have *rationally reconstructed* one of the only MT systems in daily production use: the METEO system. We show how a combination of a sentence-based memory approach, a phrase-based statistical engine and a neural-network rescorer can give results comparable to those of the current system while offering a faster development cycle and arguably better customization possibilities.

1 Introduction

Machine Translation is a field nowadays strongly anchored in a paradigm of performance. Evaluation exercises such as those conducted within the TIDES project are pushing system designers to constantly improve their systems. The *shared task* usually consists in translating news articles excerpts from a foreign language into English. While this is certainly a challenging issue, real life applications of machine translation in a production setting (i.e. without human revision) will likely be more targeted than newspaper articles.

More focused evaluation exercises do exist. Within the IWSLT workshop (Akiba et al., 2004), the main objective was to provide an evaluation framework for spoken language translation technologies. The shared task consisted in translating sentences from the Basic Travel Expression Corpus (BTEC) which gathers sentences believed to be useful for a tourist in a foreign country. In the Verbmobil project (Wahlster, 2000), transcriptions of spontaneous speech from several narrow domains such as appointment scheduling were translated from German into English.

In this study, we focus on an even more concrete task and one of the greatest successes of machine translation: the English to French translation of weather forecasts issued by Environment Canada[1], which we call here the METEO task. Machine translation of weather forecasts helps guarantee that the translations are produced in a timely manner given the very short life of a weather report. Weather reports are issued every 6 hours.

We specifically chose this task because there already exists a fully operational rule-based translation system designed for it, who's performance was carefully measured (Macklovitch (1985)), and because a very large bitext corpus of previously published weather forcasts is available. Thus, we were able to build a variety of corpus-based MT systems targeted on this specific task and compare their respective performances.

[1] The current reports are available at http://meteo.ec.gc.ca/forecast/textforecast_f.html

In the mid seventies, a group of linguists and computer scientists of Université de Montréal (*TAUM* group) developed an English to French weather report machine translation system which became known as the TAUM-METEO system. Spin-offs of that research system, namely METEO-1 and METEO-2 (Grimaila and Chandioux, 1992), have been in continuous use since 1984 translating up to 45,000 words a day. A description of the system is presented in (Hutchins and Somers, 1992, chap12). Roughly, it involves three major steps: dictionary look-up, syntactic analysis and light syntactic and morphological generation.

Leplus et al. (2004) described experiments they conducted on a bitext of forecasts in both French and English produced during 2002 and 2003 by Environment Canada. They report fairly good translation results by applying a straightforward memory-based approach to the task. The authors mentioned that this approach is warranted by the particularly high rate of sentence repetitions. Indeed, Grimaila and Chandioux (1992) argued that the repetitiveness of the task was one of the reasons for the success of the METEO system.

In the current work, we explore how well state-of-the-art corpus-based approaches can do for the same task. We have considered several techniques and their combination. In particular, we show in Section 3 that a memory-based approach of the kind described by Leplus et al. (2004) is particularly appropriate to the task. We also show in Section 4 that a statistical phrase-based engine based on the PHARAOH decoder (Koehn, 2004) also does well. In Section 5 we describe our experiments in bootstrapping translations produced by both techniques, following an approach described by Bangalore et al. (2001). In Section 6, we report the positive results we obtained in deploying a multi-layer linear perceptron to rescore the n-best lists produced by our SMT engine, an idea initially proposed by Gandrabur and Foster (2003). The global performance of our system is then analysed in Section 8.

		English		French	
bitext	sent	words	toks	words	toks
TRAIN	4 188	30 326	9.9	37 330	11.1
BLANC	122	888	3.0	1 092	3.2
TEST	36	269	1.8	333	2.0
total	4 347	31 482	10.1	38 756	11.3

Table 1: Main characteristics of the bitext used. Counts are expressed in thousands.

2 The METEO bitext

We used in this work a bitext fully described in (Leplus et al., 2004) and relied on the same splits into three subparts, namely TRAIN for training the SMT engine, BLANC for tuning purposes, and TEST that we used only at the end of the work to test our approaches. These partitions were chosen to be disjoint in time; the TEST corpus is taken from a period later than the TRAIN one, in order to simulate a real setting in which an MT system would have to translate weather reports for a period following the training one.

Figure 1 gives an example of an original English weather report and its French translation. The main characteristics of this material are reported in Table 1. In addition to its size, it is a rather unusual bitext: although it includes more than 4 million pairs of sentences produced over two years, its vocabulary is fairly small (around 10 000 words); weather forecasts are thus a very specific domain.

3 The memory-based approach

As this bitext is highly repetitive, we first started our experiments by investigating the sentence-based memory approach advocated by Leplus et al. (2004). We found that 83% of the sentences of BLANC belong to TRAIN. The introduction of a few token classes (days, months or telephone numbers) further improves the coverage to 87%.

We built a translation memory by keeping the 5 most frequent translations for each different source sentence in TRAIN. In fact, 89% of the English sentences in TRAIN have only one translation, probably because most of the target sentences have been machine translated by the ME-

SUMMARY FORECAST FOR WESTERN QUEBEC ISSUED BY ENVIRONMENT CANADA	RESUME DES PREVISIONS POUR L'OUEST DU QUEBEC EMISES PAR ENVIRONNEMENT CANADA
MONTREAL AT 4.30 PM EST MONDAY 31 DECEMBER 2001 FOR TUESDAY 01 JANUARY 2002. VARIABLE CLOUDINESS WITH FLURRIES. HIGH NEAR MINUS 7.	MONTREAL 16H30 HNE LE LUNDI 31 DECEMBRE 2001 POUR MARDI LE 01 JANVIER 2002. CIEL VARIABLE AVEC AVERSES DE NEIGE. MAX PRES DE MOINS 7.

Figure 1: An example of an English weather report and its French translation.

TEO system; we will come back to this aspect in section 8.

Formally, our memory $\mathcal{M}$ is a set of M entries p_i, each one being described by e_i, the source sentence, and the set of its k_i translations f_i^j along with their cooccurence count with e_i in the training corpus n_i^j:

$$\begin{aligned} \mathcal{M} &= \{p_1, \ldots, p_M\} \\ p_i &= \Big(e_i, \{(f_i^j, n_i^j)\}_{j \in [1,k_i]}\Big) \quad \text{where } i \in [1, M] \\ & \qquad\qquad\qquad\qquad\qquad \text{and } k_i \leq 5 \end{aligned}$$

For e, a new sentence to be translated, we seek the N closest source sentences in $\mathcal{M}$ in terms of edit distance. When there are more than N source sentences in the memory with an equal edit distance from e, we consider the most frequent ones (the approximative frequency of e_i is computed by summing the cooccurence counts n_i^j over j). Let $r = r_1, \ldots, r_N$ be the ranks of these closest entries in the memory. The ranked list of candidate translations for e is obtained by ranking each target sentence of p_{r_n} according to a score which favors first the smallest edit distances, then the relative frequency of a translation in its entry. These many translations will be combined by a technique described in Section 5.

We experimented with several other parameters, such as M, the number of entries in the memory. Although we found (on the BLANC corpus) that almost similar results could be obtained with only the 18 872[2] most frequent source sentences in TRAIN, we report results with M=488 786, the total number of different source sentences in TRAIN. While this incurs a penalty in translation time, in our implementation, looking for all translations of a given sentence takes on average less than 2 seconds on a standard desktop computer.

	WER	SER	NIST	BLEU
BLANC	8.78	23.92	11.2726	87.04
TEST	8.42	23.43	10.9571	87.68
Leplus	*9.18*	*23.56*	*10.8983*	*86.95*

Table 2: Results of the memory-based approach in terms of word error rate WER, sentence error rate SER and n-gram precision scores NIST and BLEU. The last line indicates the results reported by Leplus et al. (2004) on the same TEST corpus.

We evaluated this approach by systematically picking the best ranked translation found in the memory. We applied four standard metrics to rate the system against a single reference translation: two error rates, WER and SER and two n-gram precision scores: NIST and BLEU, both computed by the `mteval` script available from the NIST web site[3]. The results are reported in Table 2.

These results are very good compared to those observed on other translation domains, *e.g.* those described by Zens and Ney (2004) who give state-of-the-art performances in three different translation tasks, including Verbmobil. Our results are slightly better than those reported by Leplus et al. (2004), but in any case, the sentence error rate we obtained is not particularly low; especially if we recall that 87% of the source sentences of BLANC were found verbatim in TRAIN. Indeed, we found some fluctuations in the translations present in the reference. For instance, 7.2% of the source sentences in BLANC have a translation which is not the one most frequently found in the reference for these sentences. We also have to remember that these rates are mea-

[2] This happened to be the number of different sentences seen at least 10 times in TRAIN.

[3] `http://www.nist.gov/speech/tests/mt/mt2001/resource`

sured against a single reference. We give in Section 8 a more detailed analysis of the performance of the whole system.

4 The SMT approach

The second approach we investigated was to build a phrase-based statistical engine, based on the PHARAOH decoder (Koehn, 2004), for the METEO task. PHARAOH is a fast, carefully documented decoder which is easy to use, requiring a language model and a translation table; if desired, weighting coefficients as well as a few pruning options can control the behavior of the engine.

We split the TRAIN corpus in two subparts, TRAIN-T (4 180 000 pairs of sentences) for training the translation and the language models, and TRAIN-H (8 100 pairs) for tuning the different parameters of the engine. We trained a Kneser-Ney smoothed trigram language model using the SRILM package (Stolcke, 2002). The perplexity of this model on BLANC and TEST is respectively 4.94 and 3.83, which is very low compared to standard benchmarks (Zens and Ney, 2004).

To build our translation table, we first aligned our bitext at the word level. Following a common practice, we used the GIZA++ package (Och and Ney, 2000) to word-align our bitext in both directions (English-to-French and French-To-English)[4]. We extended the set of word links that were present in both alignments by some links belonging to only one alignment direction, following the heuristics described in (Koehn et al., 2003). From the resulting alignment $\mathcal{A}$, we collected the set of pairs of source and target sequences (f_a^b, e_i^j) from all regions $(a, b) \times (i, j)$ in the alignment matrix where none of the source words in f_a^b is aligned to a word not belonging to e_i^j and vice-versa:

$$\begin{array}{l} \forall x \in [a, b], \forall y : (x, y) \in \mathcal{A}, y \in [i, j] \\ \forall y \in [i, j], \forall x : (x, y) \in \mathcal{A}, x \in [a, b] \end{array}$$

We did apply a few length-based heuristics to filter the parameters acquired in this way: (source or target) sequences of at most 8 words were considered and we imposed that the length of the longest sequence in a pair was at most twice the length of its counterpart[5].

	WER	SER	NIST	BLEU
BLANC	8.17	32.46	10.4081	83.52
TEST	7.46	32.01	10.8725	84.03

Table 3: Results of the phrase-based statistical engine on the BLANC and TEST corpora.

Doing so, we acquired a model of 3 199 163 parameters. We considered two ways of scoring each parameter. The first one is by relative frequency, that is, simply by counting the number of times a given pair (f, e) was seen aligned in the bitext, normalized by the number of times f was seen. The second score we used is the IBM model 1 conditional probability (Brown et al., 1993):

$$p(e_i^j | f_a^b) = (b - a)^{-j+i-1} \prod_{y=i}^{j} \sum_{x=a}^{b} p(e_y | f_x) \quad (1)$$

Since we had only a few parameters to tune, we sought the best setting by uniformly sampling each parameter range with a small enough step size (0.1 for weighting coefficients, and 0.2 for the word penalty). Indeed, we did not find the tuning to bring much gain. In particular, a contrario to the observation we made on other translation tasks, we did not observe a huge difference in performance between the relative frequency score and the IBM one. This might be due to the fact that each parameter is seen often enough that relative frequency is sufficiently discriminative. Results of our engine on the BLANC and TEST corpora are reported in Table 3.

A direct comparison of the performance of the memory-based system and the SMT one goes in favour of the former approach (especially if we consider SER). However, the performance of the phrase-based engine on the only sentences that were not seen in the TRAIN corpus are much better (1 point more of BLEU% score)

[4] We used the alignments produced by IBM model 2.

[5] We played with all the heuristics without noticing significant impact on performance.

5 Bootstrapping experiments

Bangalore et al. (2001) have shown, on a domain-dependent spoken dialogue translation task, that combining the output of several off-the-shelf translation engines resulted in better performance than the one of each individual engine. Similar results were reported on a more general domain translation task in (Bangalore et al., 2002). The key underlying idea is to use the word alignment of the output of different translation engines in order to identify the locus of consensus.

We followed the approach described by Bangalore et al. (2002) and adapted the CLUSTALW multiple-string aligner first designed for biological sequence alignment (Thompson et al., 1994) to our domain[6]. An example of multiple-sequence alignment from the $N = 10$ best ranked translations output by the memory-based system on a single translation session is given in Figure 2. In this example, no candidate translation agreed with the reference on every single word, but it is often the case that most of them agree on some units such as `MAXIMUM DE 12` (*high of 12*) ou `TOT CE MATIN` (*early this morning*).

We then built a lattice out of this alignment that can generate both the produced translations as well as new ones. The lattice corresponding to the previous example is given in Figure 4. Using the CARMEL package (Knight and Al-Onaizan, 1999), we found a lowest cost path in this automata in order to produce a final translation. The 10 lowest cost consensus translations produced out of the ones reported in Figure 4 are indicated in Figure 3. This example shows the tendency of the consensus translations to be more consistent with each other than were the ones provided by the memory. This is also the trend we observed by casual inspections of the consensus translations we produced over the BLANC corpus.

We tried several variations on this idea. We first considered different ways of weighting an arc of the lattice, using various combinations of the native probability of the automaton, and the probability provided by a language model trained on the full target side of the TRAIN material. None of the experiments we conducted with the language model yielded satisfactory results. This might be explained by the fact that it is a too general model for discriminating between specific sentences. We finally scored each arc with the native counts obtained at construction time, giving it a credit inversely proportional to the first rank in the nbest-list where the transition sequence is observed.

We also investigated bootstrapping of the translations drawn from the memory, from the SMT engine and from both of them, but observed positive results in the first case only. These are reported in Table 4 for the only source sentences that were not found verbatim in the TRAIN corpus. As indicated by the different metrics, translation by consensus improves the overall quality of the output of the memory. Sentence error-rate decreased by almost 10 points, a substantial im-

Source `HIGH 12 EARLY THIS MORNING .`
Reference `MAXIMUM 12 TOT CE MATIN .`

```
MAXIMUM DE             12         CE  MATIN      .
MAXIMUM                12 ATTEINT CE  MATIN      .
MAXIMUM DE             12 TOT     CET APRES-MIDI .
MAXIMUM DE  PLUS       12 TOT     CE  MATIN      .
NAPPES  DE  BROUILLARD    TOT     CE  MATIN      .
BRUMEUX PAR ENDROITS      TOT     CE  MATIN      .
MAXIMUM DE             12         EN  MATINEE    .
BRUMEUX                   TOT     CE  MATIN      .
MAXIMUM DE  PLUS       12         CE  MATIN      .
MAXIMUM DE  MOINS      12         CE  MATIN      .
```

Figure 2: Multiple-sequence alignment from the ten best-ranked translations provided by the memory-based system for the source sentence: `HIGH 12 EARLY THIS MORNING.`

Source `HIGH 12 EARLY THIS MORNING .`
Reference `MAXIMUM 12 TOT CE MATIN .`

```
MAXIMUM DE PLUS 12 CE MATIN .
MAXIMUM DE 12 CE MATIN .
MAXIMUM DE PLUS 12 TOT CE MATIN .
MAXIMUM DE 12 TOT CE MATIN .
MAXIMUM DE TOT CE MATIN .
MAXIMUM DE ENDROITS TOT CE MATIN .
MAXIMUM DE BROUILLARD TOT CE MATIN .
MAXIMUM DE MOINS 12 CE MATIN .
MAXIMUM DE PLUS 12 EN MATINEE .
MAXIMUM DE PLUS 12 ATTEINT CE MATIN .
```

Figure 3: Translations after the consensus from the sentences reported in Figure 2.

[6] This meant extending the number of different symbols that could be aligned by CLUSTALW and modifying the cost matrix.

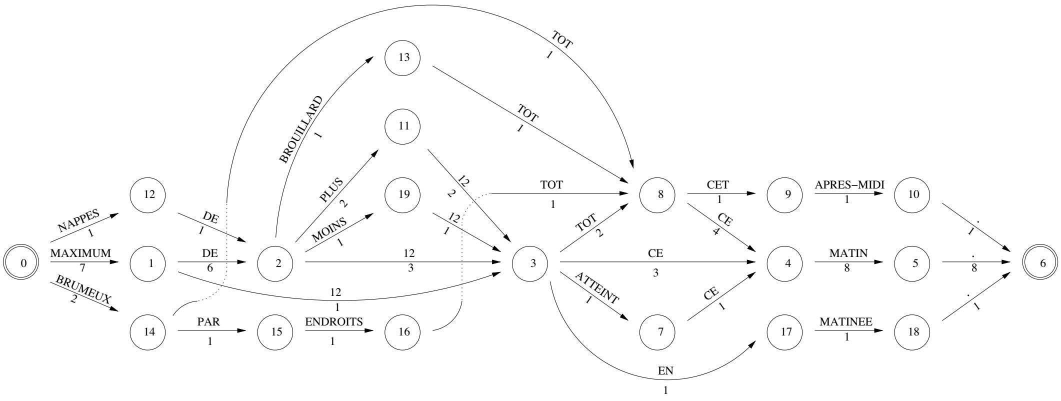

Figure 4: Lattice produced for the translations of Figure 2. The weights on the arcs are the frequency of a given transition. A non-smoothed local bigram language model is obtained by simply normalizing each node by the sum of the weights of the arcs leaving this node.

provement.

	WER	SER	NIST	BLEU
memory	18.69	94.82	9.7853	66.56
+ consensus	18.97	85.53	9.9314	68.86

Table 4: Results of the consensus approach on the output of the memory, for the 13 010 sentences of BLANC not seen verbatim in the TRAIN corpus.

6 The rescoring approach

Several papers recently described *rescoring* approaches for improving the accuracy of a base MT system (*e.g.* (Blatz et al., 2004)). Rescoring is based on the hope that with the help of additional information (or different ways of using it), we can change the ranks of translations in a n-best list. One motivation for such an approach is that the best translation among alternatives is often not the first one proposed by the base system.

PHARAOH, used in section 4, can output its search graph for which the CARMEL package (Knight and Al-Onaizan, 1999) used in the preceding section, can produce n-best lists. For each source sentence s we built a n-best list of up to 10 different translation alternatives $\{t_j\}_{j\in[1,n\leq 10]}$ using the phrase-based model described above. Each translation alternative, represented as a vector v_j of feature functions and tagged as either correct $\oplus$ or wrong $\ominus$, constitutes a *rescoring example*. A translation was tagged as correct if and only if it was identical to the corresponding reference translation.

The rescoring model we used was a multi-layer perceptron (MLP) with a LogSoftMax activation function, trained by gradient descent with a negative-log-likelihood criterion. With this setup, the MLP is trained to estimate $p(\oplus|v_j)$, the conditional probability of correctness of each candidate translation t_j. We experimented with different numbers of hidden units within one single hidden layer and found the best results (on the validation set) with 25 hidden units. All MLP experiments were done using the open-source machine learning library Torch (Collobert et al., 2002).

For training and validation of the MLP, we used examples extracted from BLANC: out of a total 872 249 training examples, we kept 8 000 for validation purposes. The testing of the rescoring MLP was performed on the TEST corpus, which represents 261 577 testing examples.

Note that for each translation alternative t_j the base system actually can produce more than one decoding hypothesis h^i_j, depending on how it segmented t_j into chunks produced by the phrase-based model. Each such segmentation returns a different native probability estimate p^i_j.

The rescoring feature functions we used are:

- the ratio of the length of s over the length of

	WER	SER	NIST	BLEU
baseline	7.46	32.01	10.8725	84.03
rescoring	5.73	25.03	10.9828	87.40
oracle	2.42	14.10	11.5082	92.33
baseline	14.63	66.54	10.5912	76.04
rescoring	13.71	64.48	10.6968	77.45
oracle	8.58	48.82	11.3217	83.81

Table 5: Translation accuracy obtained with the native phrase-based system (baseline), with the rescoring system (rescore) and with the WER oracle (oracle), that is, the best performance we could have with respect to WER. The first half of the table is for the full TEST corpus, while the second half is computed over the subset of the sentences of TEST that do not occur in TRAIN.

t_j: For a given pair of languages, this ratio is usually homogeneous;

- from the decoding hypothesis $h^r_j = \mathrm{argmax}_i(p^i_j)$ that has the highest native score among hypotheses h^i_j corresponding to t_j, we retained the native score p^r_j as well as different statistics on chunk size. Longer chunks appear when the translation resembles a reference translation.

- the posterior probability estimate $c(t_j)$ captures the frequency of a translation t_j weighted by the native scores of all its corresponding decoding hypothesis h^i_j:

$$c(t_j) = \frac{\sum_i p^i_j}{\sum_j \sum_i p^i_j}$$

  This feature is more significant than the frequency or the native score taken in isolation and is a sound normalization that makes it independent of the sentence length.

- the score of the IBM model 1 and model 2 normalized by the length of t_j: These turned out to be the most significant features (model 2 slightly better than model 1), as reported in (Och et al., 2004): Word-based SMT models complement well the information captured by phrase-based SMT models.

Table 5 shows the gain in translation accuracy obtained by the rescoring layer. The base-line corresponds to choosing the translation alternative with the highest native score. WER-oracle, the alternative with the lowest WER, is the optimal translation within the n-best set. Rescoring significantly improved translation accuracy over the TEST corpus as reflected by all four evaluation metrics.

Improvement is less important over the subset of “hard” sentences of TEST, i.e. the sentences not in the TRAIN corpus. This may be caused by our tagging procedure: we tag all translation alternatives that have one or more word-errors with respect to the reference as false. This means that the rescoring MLPs was trained to detect correct translations (with WER= 0) but wasn’t optimized to rank “bad” translations according to their WER. Thus, on the “hard” test corpus, where most of the translations are “bad”, the MLPs are less accurate.

7 Combination

We have shown that statistical phrase-based translation is better at predicting new sentences than the translation memory alone. This suggests a simple combination scheme of translation memory and SMT engine: full sentences found in the memory are retrieved from the memory verbatim and the others are translated using the rescored phrase-based SMT system. This combined set-up yielded in fact the best results, as shown in table 6.

8 Discussion

We have compared various ways of setting up corpus-based approaches for a well-defined real life task: the translation of meteorological re-

	WER	SER	NIST	BLEU
memory	8.42	23.92	11.2726	87.04
rescoring	5.73	25.03	10.9800	87.40
combination	4.85	20.80	11.3021	89.59

Table 6: Performance on the TEST corpus of a simple combination of the translation memory system and the rescored phrase-based SMT engine.

ports. The advantage of this application is both the availability of huge amounts of bitexts and the existence of a commercially used rule-based MT system for the task. Unfortunately, we could not access the rule-based system directly, so we had to rely on its published outputs to infer its results; and even there, we had no indication of the level of revision done on the raw output of the system.

We built a series of task-specific corpus-based MT systems, ranging from translation memories to phrase-based SMT with neural net rescoring using word-based models. We observed that a straightforward memory-based approach can already obtain good results, thanks to the highly repetitive nature of the weather forecast domain. We found that a phrase-based SMT engine is even better suited to translate previously unseen sentences. We also reported a further improvement after applying a rescoring layer. Finally, combining both systems yielded significant overall improvements.

While these results are encouraging, we might still wonder how they compare to actual performance of the METEO system. The only carefully described evaluation of METEO we could find is the one the Translation Bureau conducted on the METEO 2 system twenty years ago (Macklovitch, 1985). We have good reasons to believe that the system has not changed substantially since then except for an update of its dictionaries and its computing infrastructure. In this study, Macklovitch sampled a set of 1257 sentences produced over a 24-hour period by *Environnement Canada*. He counted the number of times the machine translation was identical to the final revised version. However, he took care to remove those errors that arose as a result of typos or clear omissions in the original source (English) text. He found that only 11% of the sampled sentences were different from the revised ones.

This evaluation setting roughly corresponds to the SER. Macklovitch also reports that a requirement for the METEO system then was that at least 80% of the sentences submitted to the system should be translated without any human post-editing.

While the corpus-based approaches we developed almost meet this last requirement, we must admit that the sentence error rates we measured are still higher than the one Macklovitch measured on the METEO2 system. This might look at first a bit disappointing, but this comparison must be taken with a grain of salt. First, our evaluation was conducted over a much larger test set (36 228 sentences in our case). Second, we observed that 7% of the translations of English sentences found in the memory did not match their single reference translation. This suggests that the SER we measured are actually upper-bound rates, since those *bad* translations are in fact good (and duly revised) ones. Third, an informal evaluation on a random sample of translations that differed from the reference showed that 77% of these *bad* translations were found acceptable by humans.

But even if we had outperformed the current system, it would not mean that the weather bureau would migrate from it. We have been discussing recently with people from Environment Canada and they have suggested that we try our translation approach on the weather alerts which are issued (almost) daily and that are currently being translated by humans because the current system cannot deal with them. Given the urgency of this information for the public, the alerts must be broadcast rapidly and Environment Canada is looking for ways of speeding up the delivery of this information in both French and English. They have provided us with five years of different types of weather alerts and we are currently investigating how well the approaches we presented here are appropriate for translating this material. We could expect that there is less repetition in these special kind of reports but given the good success we had with our statistical engine, we are confident that we can rapidly develop an efficient tool for this task.

Some alternatives to Machine Translation (MT) have been proposed for weather reports, namely multilingual text generation directly from raw weather data: temperatures, winds, pressures etc. In these generation systems, humans still make selections on templates in order to organize the report. Generating text in many languages from one source is quite appealing from a conceptual point of view and has been cited as one of the potential applications for natural language generation (Reiter and Dale, 2000); some systems have been developed (Kittredge et al., 1986; Goldberg et al., 1994; Coch, 1998) and tested in operational contexts. A recent experiment (Reiter, 2005) even suggests that in some ways automatically generated reports are judged better by humans than human written forecasts! But to our knowledge, no automatic generation has yet been put to daily use on the same level as the one attained by MT, one of the reasons being that meteorologists prefer to write their reports in natural language rather than selecting text structure templates. So we are confident that there is still a need (and a market ?) for a the automatic translation of manually written weather reports.

9 Acknowledgements

We thank Rick Jones and Marc Besner from the Dorval office of Environment Canada who provided us with the corpus of bilingual weather reports. We are also indebted to Elliot Macklovitch who provided us with articles describing the METEO system that we could not have found otherwise. This work has been granted by NSERC and FQRNT.

References

Y. Akiba, M. Federico, N. Kando, H. Nakaiwa, M. Paul, and J. Tsujii. 2004. Overview of the IWSLT04 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 1–12, Kyoto, Japan.

S. Bangalore, G. Bordel, and G. Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *Proceedings of ASRU*, pages 351–354, Trento, Italy, Dec.

S. Bangalore, V. Murdock, and G. Riccardi. 2002. Bootstrapping bilingual data using consensus translation for a multilingual instant messaging system. In *Proceedings of COLING*, pages 50–56, Taipei, Taiwan.

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., Ueffing, and N. 2004. Confidence estimation for machine translation. In *Proceedings of COLING 2004*, pages 315–321, Geneva, aug.

P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

J. Coch. 1998. Interactive generation and knowledge administration in multimeteo. In *Ninth International Workshop on Natural Language Generation*, pages 300–303, Niagara-on-the-lake, Ontario, Canada.

R. Collobert, S. Bengio, and J. Mariéthoz. 2002. Torch: a modular machine learning software library. Technical Report IDIAP-RR 02-46, IDIAP.

S. Gandrabur and G. Foster. 2003. Confidence estimation for text prediction. In *Proceedings of CoNLL*, Edmonton, may.

E. Goldberg, N. Driedger, and R. Kittredge. 1994. Using natural language processing to produce weather forecasts. *IEEE Expert 9*, 2:45–53, apr.

A. Grimaila and J. Chandioux, 1992. *Made to measure solutions*, chapter 3, pages 33–45. J. Newton, ed., Computers in Translation: A Practical Appraisal, Routledge, London.

W.J. Hutchins and H.L. Somers. 1992. *An Introduction to Machine Translation*. Academic Press.

R. Kittredge, A. Polguère, and E. Goldberg. 1986. Synthesizing weather reports from formatted data. In *11th. International Conference on Computational Linguistics*, pages 563–565, Bonn, Germany.

K. Knight and Y. Al-Onaizan, 1999. *A Primer on Finite-State Software for Natural Language Processing*, August. http://www.isi.edu/licensed-sw/carmel/carmel-tutorial2.pdf.

P. Koehn, F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT*, pages 127–133.

P. Koehn. 2004. Pharaoh: a beam search decoder for phrase-based smt. In *Proceedings of AMTA*, pages 115–124.

T. Leplus, P. Langlais, and G. Lapalme. 2004. Weather report translation using a translation memory. In *Proceedings of the 6th AMTA*, pages 154–163, Washington DC, USA, September.

E. Macklovitch. 1985. A linguistic performance evaluation of meteo 2. Technical report, Canadian Translation Bureau, Aug.

F.J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL*, pages 440–447, Hongkong, China.

F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of HLT/NAACL 2004*.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press. 270 p.

E. Reiter. 2005. Summary of sumtime weather forecast experiment.

A. Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of ICSLP*, Denver, Colorado, Sept.

J.D. Thompson, D.G. Higgins, and T.J. Gibson. 1994. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680.

Wahlster, editor. 2000. *Verbmobil: Foundations of speech-to-speech translations*. Springer Verlag, Berlin, Germany, July.

R. Zens and H. Ney. 2004. Improvements in phrase-based statistical machine translation. In *Proceedings of HLT/NAACL*, pages 257–264, Boston, MA, May.