DEMOCRAT: Deciding between Multiple Outputs Created by Automatic Translation

Menno van Zaanen
Centre for Language Technology
Macquarie University
North Ryde NSW 2109, Australia
menno@ics.mq.edu.au

Harold Somers
School of Informatics
University of Manchester
Manchester M60 1QD, UK
harold.somers@manchester.ac.uk

Abstract

We present DEMOCRAT, a system which DEcides between Multiple Outputs CReated by Automatic Translation, specifically free on-line MT systems, and tries to construct from them a consensus translation which, it is hoped, will take the best elements of the contributing systems and produce an output as good as or better than any of the individual MT systems on their own. We review the small number of previous implementations of this variation of multi-engine MT, contrasting our own algorithm with those reported elsewhere. Other implementations all use language models and require extensive training, whereas DEMOCRAT is "plug-and-play". Also, they have been evaluated only on single language pairs and texts, whereas we have experimented with a larger variety, and found the results to be variable, but consistent. We therefore consider what factors contribute to DEMOCRAT's performance.

1 Introduction

The number of competing Machine Translation (MT) systems is continually growing, with many of them available free on the Internet. One can quickly observe that while some are noticeably better than others, it is not generally the case that there is a single outstanding system that always gives the best results: rather, most of them give good results some of the time. It is therefore an interesting goal to take multiple outputs from different systems and to attempt to combine the best of them to produce output that is better than any of the individual contributing systems and consistent over more types of text. In this paper we present DEMOCRAT, a system which DEcides between Multiple Outputs CReated by Automatic Translation.

The idea of applying multiple MT systems and reconciling their outputs was first suggested by Frederking and Nirenburg (1994), and was given the name "multi-engine MT" (MEMT). Significantly, the different engines were all developed in-house. The idea to combine outputs from multiple off-the-shelf MT systems seems to have been first suggested by Bangalore et al. (2001, 2002), inspired by the positive effects of combining outputs from multiple systems in other applications such as text categorisation (Larkey and Croft, 1996), speech recognition (Fiscus, 1997) and POS tagging (Roth and Zelenko, 1998). Apparently independently, Nomoto (2003, 2004) proposed something similar. Jayaraman and Lavie (2005) represent the only other work on this idea known to us. All of the above tackle the problem in different ways, as do we; but all reach the same conclusion, that combining the outputs results in a better translation, however judged, than any of the individual contributing outputs. As Frederking and Nirenburg (1994) put it, "three [or more] heads are better than one".

2 Related work

2.1 MEMT

The multi-engine approach to MT was pioneered by Frederking and Nirenburg (1994). They pass the output from three independent MT architectures – KBMT, EBMT and a lexical transfer system – onto a chart, which is then traversed to produce a single output. Crucial to the "chart walk" algorithm is that each output comes with a confidence score computed by the individual engine. The system was further developed Brown and Frederking (1995) by the addition of an n-gram-based mechanism for candidate selection. A similar architecture

has been used for a variety of language pairs in the DIPLOMAT system (Frederking et al., 1997), in Nyberg and Mitamura's (1997) system for translating captions, in Rayner and Carter's (1997) SLT system, in Akiba et al.'s (2002) ATR system, and in Lavie et al.'s (2003) system.

2.2 Consensus translation

Unlike the MEMT approaches mentioned above, our work fits into a smaller paradigm of studies which try to produce "consensus translations" (Bangalore et al., 2002) from off-the-shelf MT systems which are thus "black boxes" (Nomoto, 2003). As Nomoto says, traditional MEMT design is "based on the knowledge it has about [the] inner workings of each of the component engines" (p. 269). If you want to add a new MT engine, the scoring mechanism has to be redesigned. In our approach, the only information available is the raw source text and its equally raw translation(s).

Nomoto's method nevertheless requires something like the confidence scores found in the original MEMT, and he develops statistical models which are used to measure the perplexity of the competing outputs. These are based on the IBM models of Brown et al. (1993) and are trained with a support vector regression technique using three parallel and two monolingual corpora, some of them of considerable volume.

Bangalore et al. (2001, 2002) use a simpler method, rather like the "chart walk" of the original MEMT. The multiple outputs are first aligned using a "progressive multiple alignment" technique found in the biological sciences. The output is represented as a lattice in which parallel arcs represent alternative translations, with associated weights reflecting whether alternatives were found in more than one output. A least-cost traversal of the lattice corresponds to selecting the consensus [translation] by majority vote (CMV). Where there is no clear majority, Bangalore et al. employ a simple n-gram language model based on a moderately large corpus (58,000 sentences) of translations using the same MT systems. The CMV method alone is as good as the best of the MT systems, and the addition of the language model provides translations consistently better than the contributing MT systems.

Jayaraman and Lavie (2005) first align the contributing outputs using a basic edit distance ignoring case, and using a stemmer to increase the number of matches. Strings are then combined iteratively, under certain conditions, to produce a set of hypotheses. Like the other systems, a third stage involves ranking the hypotheses according to a combination of a "standard trigram language model trained on large corpora of the target text", and confidence scores for each word "associated with the system which produced it". Various ways of setting the confidence scores are mentioned, all requiring a calibration phase which could be quite lengthy.

3 Method

Our approach differs from previous work on this idea in that we wished to experiment with an algorithm that requires nothing more than the raw outputs provided by the MT systems: if it works for a variety of language pairs, it would be instantly applicable to any situation where multiple alternative MT outputs are available. All the other approaches to this problem so far reported include an element of training the system on language models: although they are often simple n-gram models, they nevertheless require sometimes considerable amounts of previously translated text. Two of them (Nomoto, and Jayaraman and Lavie) also require confidence scores associated with each of the outputs.

Our approach also differs in the way it actually chooses from among the alternative translations. Like Bangalore et al. and Jayaraman and Lavie (but not Nomoto), our system incorporates an attempt to align the multiple inputs. Unlike Jayaraman and Lavie whose alignment algorithm includes a stemmer, our system does not rely on any pre-processing whatsoever. We consider each word as is, with no conversion to lowercase, stemming, or other techniques applied beforehand.[1] Our system thus remains language-independent. DEMOCRAT differs from Bangalore et al.'s system in the way alternative choices are weighted: their system uses probability information from a target-language corpus, while ours works only on the given outputs.

The main idea behind DEMOCRAT is to anal-

[1]Except that with writing systems that do not indicate word boundaries, such as Chinese and Japanese, we will need to incorporate a segmentation stage.

yse the translations output by the systems and select and combine the "best" sequence of words from them to give one consensus translation. We assume that if many MT systems use the same word or phrase, it is likely to be good. As the name of the system suggests, we select single words based on a majority vote of the output of the systems.

The algorithm consists of three distinct phases. The first phase aligns all the input sentences in pairs. The second phase analyses the alignments and builds a graph that contains the input sentences combined with the information on words that are shared. The third and final phase walks through the graph and generates the output sentence.

The first phase aligns each pair of input sentences using the edit-distance algorithm (Wagner and Fischer, 1974), which computes the Levenshtein distance (Levenshtein, 1965), i.e. the smallest number of edit operations (insertion, deletion and substitution, with corresponding costs of 1, 1, and 2) needed to convert one string into another. We are not so much interested in the actual distance, but in the alignment that is also computed.

The process can be illustrated with the three outputs from MT systems in (1)–(3).

(1) Approval of the official report of the preceding meeting

(2) Approval of the verbal process of the preceding meeting

(3) Approbation of the minutes of the previous session

An initial graph is created as follows. From a unique start node we create edges to sequences of nodes representing each output. The nodes labelled with the last word of each output are linked to a single terminal node. Figure 1 gives an example of the initial graph for the three outputs in (1)–(3).

Based on the pairwise alignments found between each pair of outputs, this graph is now compressed. Where a match is found between two words in the two outputs, the two separate nodes in the initial graph are merged into one. This is done for each pair of outputs, until the graph is completely compressed. For each merged node (and corresponding edges), frequency information, i.e. the number of paths through that node, is recorded. This provides us with information on how well used that particular subpath is. In fact, this will drive the last phase of the algorithm. The result of this process for our example is depicted in figure 2.

Note that all original outputs are still present in this graph and can be generated by a particular path through the graph. However, alternative outputs can also be generated when other paths are followed. For example, *Approbation of the official report of the preceding meeting* can be generated, although it was not one of the phrases we started out with.

The final phase starts from the start node and selects the "best" next node. This selection is based on the frequency that is stored with the edge and node. In the example, *Approval* would be a better choice than *Approbation*, as two MT systems generated this word (as the frequency information indicates). From there, *of the* is the only choice. When there is no single edge with the highest frequency (and thus no majority vote is available), currently one edge is chosen in a pseudo-random manner based on some idiosyncrasies of the Perl implementation (see discussion on improving this below). This process is repeated until the end node is found and the sentence corresponding to the path through the graph is output.

Care needs to be taken with respect to cycles. Consider the hypothetical graph in figure 3. Here we have used dashed lines to indicate nodes that are actually merged. In this case, there is a cycle in the graph, which allows strings of infinite length to be generated, e.g. $w_1, w_2, w_3, w_4, w_1, w_2, \ldots$. To prevent this, we do not allow nodes to be visited twice in the "graph walk". In this particular case, the transition from w_4 to w_1 is not allowed, as the w_1 node has already been used.

4 Evaluation

The experiments of Bangalore et al. (2001, 2002) focused on spoken language English–Spanish translation. In their 2001 article they constructed consensus translations of 1,044 sentences using five on-line MT systems. Translations were subjectively evaluated on a 3-point scale by two native speakers. A different set of 300 translations was evaluated objectively using a pairwise edit-distance metric. The 2002 article mentions evaluation of 300 sentences again

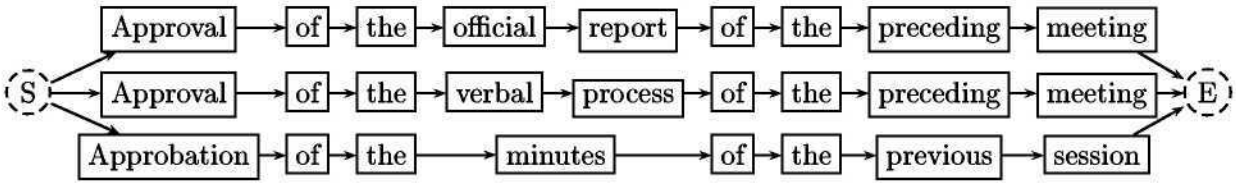

Figure 1: Initial graph of three outputs

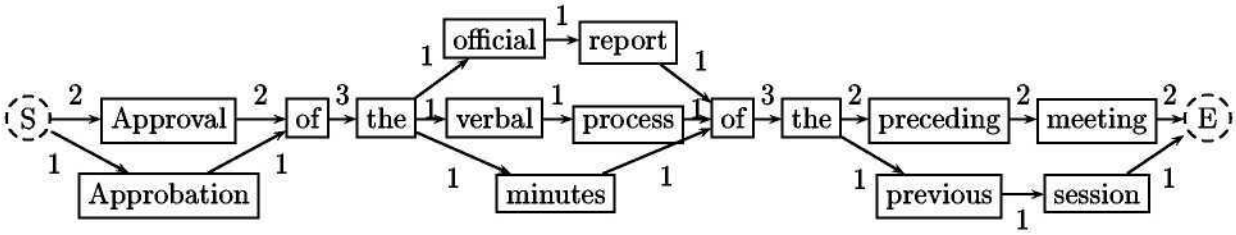

Figure 2: Result of compression of figure 1

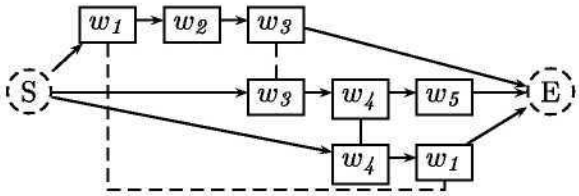

Figure 3: Cycle in a compressed graph

using an edit-distance-based metric.

The experiments reported by Nomoto (2003, 2004) involve English–Japanese translations of 10,965 examples of phrases from business letters divided into 22 blocks, and 8,307 sentences from a newspaper domain, using four on-line MT systems. The results were evaluated using a simplified variant of the well-known BLEU metric (Papineni et al., 2002).

Jayaraman and Lavie (2005) combined the outputs of three Chinese–English MT systems on three test sets each of roughly 900 sentences of newswire text. The translations were evaluated using the authors' own BLEU-like METEOR metric.

All three previous studies have found that their consensus translations, however arrived at, are as good as or better than any single one of the translations which contribute. None of the previous studies has experimented extensively with different text types, language pairs or combinations of MT systems however. Having set up DEMOCRAT to be as nearly a "plug-and-play" system as possible, we can run a large number of evaluations with various language pairs and combinations of MT systems. In this paper we focus on results of three French–English texts and one English–German translation, though we have also experimented with other language pairs, and have got similar results.

We evaluated our results using the familiar BLEU metric, and also an F-score metric devised by Turian et al. (2003), which has the advantage of separating precision and recall elements of the evaluation.[2]

As is well known, several online MT services use the same or essentially similar MT engines. In particular, Systran is well represented for most language pairs available. In our experiments we investigate the impact of multiple use of similar translations, which shows an obvious effect on DEMOCRAT, as will be discussed below. In a similar vein, it is recognised that some online MT systems deliver very poor quality translations. We rejected use of any systems which were evidently word-for-word dictionary lookup systems, but, again we will see below to what extent including poor output affects DEMOCRAT's performance.

4.1 The texts and systems

English/French is probably the language pair with the biggest choice of on-line MT systems. We evaluated DEMOCRAT on three texts translating from French into English: extracts from the web pages of the Marseilles Tourist Of-

[2]Several implementations of the BLEU metric are available on the web, giving different results (not just different scores, but ranking outputs differently). We therefore preferred to use a new implementation based on the original description, along with Turian et al.'s (2003) F-score metric, both provided by our colleague Simon Zwarts, to whom we are most grateful.

fice (mar), a passage from the Europarl corpus of European Parliament Proceedings 1996–2003 (euFE), and selections from Jules Verne's *20000 lieues sous les mers* (jv); and on one English–German translation again from the Europarl corpus (euEG). Table 1 summarises the characteristics of these texts. See Appendix for URLs.

Abbr.	Source	No. of sent.	Av. sent. length
mar	Web site	94	19.33
euFE	Europarl corpus	107	27.73
jv	Project Gutenberg	200	23.00
euEG	Europarl corpus	101	22.09

Table 1: The texts used in the experiments

In all cases we ran DEMOCRAT with input from five systems, namely Babelfish, Freetranslation, Systran, TranslateRU (ProMT), and Worldlingo.[3] We first collected the five translations for each text, and then ran DEMOCRAT with all 25 combinations of systems.

4.2 Results

Table 2 shows all our results. Any shaded cell below the line is a good result for DEMOCRAT. All three previous studies reported that their systems always got a better evaluation score than the best of the contributing individual MT systems. This was true in our case whichever of the two evaluation measures we used, in all but one case (jv with F-score). The result is particularly good according to the BLEU score with the English–German translation where 12 of the 25 DEMOCRAT combinations do better than the best MT system.

All previous work compared only the performance of the combination using all systems with that of the single systems. Here, we evaluate all possible combinations of systems separately. This provides insight into how DEMOCRAT responds to similar or low quality input data.

5 Discussion

Not all DEMOCRAT combinations beat the best individual system, and de.25, incorporating all five inputs, never does. This indicates that DEMOCRAT is quite sensitive to the quality of its inputs. Close inspection shows that, not surprisingly, DEMOCRAT does best when the contributing inputs are of good quality, but poor quality input can degrade its performance disproportionately. At first sight this looks like a bad result for us: previous studies have reported that their consensus MT system always does better than the best individual system, but we cannot make this claim (and below we discuss why this is the case). DEMOCRAT works best when it is fed with the better input. But this undermines the basic idea of using DEMOCRAT, which assumes that you do not know which is the best system.

In fact our results are more promising than this rather negative conclusion would indicate. Notice that the best MT system is not always the same one: obviously if it was, all of this work would be pointless. Even if DEMOCRAT does not always win, it is almost always in the top two or three, and it is rarely worse than the worst system. Out of 200 BLEU and F-scores, only 7 show DEMOCRAT doing worse than any of its components, all cases where there are only two inputs (and so the random choice factor is maximised), 4 of these in the de.9 row. DEMOCRAT in general creates consistent results over different language pairs and text types. So if you do not know which system is the best, DEMOCRAT is a good way to hedge your bets.

Another thing to mention is that, as its name implies, DEMOCRAT is only as good as the systems that contribute to it. As mentioned above, many on-line MT systems are actually derived from the same underlying engine: Systran is particularly well represented among free online systems, under various banners. If all the contributing outputs are already quite good, DEMOCRAT cannot necessarily improve on them, since it uses the output of these systems as input. We found that when we ran DEMOCRAT with several systems derived from the same underlying engine, it was rarely able to come first in the evaluation, although the scores were often very close. This is logical: if almost everyone is voting for the same candidate, we cannot do better than reflect that choice. Similarly, if the majority of systems happen to make a bad choice, as sometimes happens, DEMOCRAT will of course also make that bad choice (just like in real life). So one way to improve DEMOCRAT's performance is to exclude exceptionally bad systems, which can be recognised with the naked eye by most users (assuming they are translating into their own language)

[3] See Appendix for details. The systems are referred to as ba, fr, sy, tr, and wo in the results below.

System	mar		euFE		jv		euEG	
	BLEU	F-score	BLEU	F-score	BLEU	F-score	BLEU	F-score
ba	0.23033	0.25396	0.17827	0.21214	0.19091	0.22997	0.10611	0.17575
fr	0.15087	0.21803	0.14106	0.18859	0.15175	0.20908	0.10233	0.17121
sy	0.22111	0.25466	0.18277	0.21428	0.18866	0.23053	0.10385	0.17491
tr	0.18683	0.22940	0.18458	0.21511	0.15342	0.21159	0.10789	0.18274
wo	0.22732	0.25223	0.17889	0.21266	0.16674	0.21711	0.10334	0.17141
de.0 ba fr	0.18927	0.23790	0.16157	0.20141	0.17298	0.22086	0.10543	0.17321
de.1 ba sy	0.22317	0.25386	0.18047	0.21331	0.19292	0.23043	0.10692	0.17591
de.2 ba tr	0.19340	0.23756	0.18502	0.21566	0.16554	0.21768	0.10770	0.18293
de.3 ba wo	0.23077	0.25456	0.17900	0.21281	0.17859	0.22398	0.10618	0.17446
de.4 fr sy	0.18416	0.23982	0.15939	0.20015	0.17439	0.22165	0.10462	0.17308
de.5 fr tr	0.16635	0.22592	0.16753	0.20498	0.15407	0.21144	0.10912	0.17710
de.6 fr wo	0.19167	0.23835	0.15468	0.19782	0.16025	0.21409	0.09590	0.16868
de.7 sy tr	0.18917	0.23616	0.18548	0.21662	0.16383	0.21984	0.11104	0.18454
de.8 sy wo	0.22784	0.25515	0.18093	0.21361	0.17805	0.22553	0.10606	0.17416
de.9 tr wo	0.19057	0.23832	0.17476	0.21047	0.14702	0.21134	0.11022	0.17615
de.10 ba fr sy	0.20976	0.25220	0.17922	0.21242	0.18745	0.22897	0.10711	0.17679
de.11 ba fr tr	0.19772	0.24646	0.17294	0.21063	0.17029	0.22325	0.11145	0.17975
de.12 ba fr wo	0.21441	0.25016	0.17801	0.21191	0.18054	0.22629	0.10477	0.17478
de.13 ba sy tr	0.21208	0.25171	0.18307	0.21457	0.17910	0.22596	0.10865	0.17789
de.14 ba sy wo	0.22935	0.25464	0.17859	0.21255	0.18238	0.22730	0.10764	0.17465
de.15 ba tr wo	0.21503	0.25104	0.18132	0.21340	0.16400	0.22010	0.11196	0.17924
de.16 fr sy tr	0.20043	0.25059	0.17396	0.21131	0.16894	0.22189	0.12255	0.18616
de.17 fr sy wo	0.20571	0.25168	0.17771	0.21222	0.17599	0.22682	0.10628	0.17559
de.18 fr tr wo	0.19162	0.24397	0.16883	0.20987	0.16436	0.22013	0.11727	0.18268
de.19 sy tr wo	0.21011	0.24921	0.18264	0.21435	0.15892	0.22007	0.11017	0.17828
de.20 ba fr sy tr	0.20793	0.25511	0.17255	0.21217	0.17419	0.22481	0.11042	0.18056
de.21 ba fr sy wo	0.21576	0.25234	0.17574	0.21119	0.18048	0.22811	0.10402	0.17486
de.22 ba fr tr wo	0.20877	0.25369	0.16949	0.21100	0.16947	0.22451	0.10469	0.17757
de.23 ba sy tr wo	0.20977	0.25025	0.17824	0.21230	0.16732	0.22312	0.11172	0.17874
de.24 fr sy tr wo	0.20994	0.25536	0.17665	0.21360	0.16803	0.22504	0.11389	0.18366
de.25 ba fr sy tr wo	0.20337	0.25014	0.17704	0.21328	0.16952	0.22554	0.10549	0.17749

Table 2: The full results. In each column, the dark-shaded cell shows the best score for that text. Light-shaded cells above the line show the best individual on-line system, below the line any version of DEMOCRAT which beat the best system.

without recourse to reference translations and evaluation metrics.

5.1 Future work

The main difference between our results and those of our predecessors is that DEMOCRAT does not always do better than the best individual system. And an important difference between our system and all the others is that we do not include any n-gram modelling or corpus-based weighting of the inputs, nor do we do any normalisation of the texts, to help the alignment phase. In fact, no training has to be performed at all. DEMOCRAT can be used on multiple MT system outputs independent of the language, straight away.

As mentioned above, if any of the translations are particularly poor, this can adversely affect our results. The reason for this is that if all the inputs are different, the current implementation of DEMOCRAT chooses from among them more or less at random, and may of course therefore choose the output from the worst system. One way to overcome this would be to allow DEMOCRAT to learn, as it goes along, which are the better systems: by keeping track of which MT systems' inputs get used more often, the system could allow this "history" to affect its judgement and when faced with an equal choice could favour the systems which have already contributed most. This simple form of learning would have to be on a text-by-text basis (or some sort of discarding of history should be incorporated), which would have the advantage of allowing DEMOCRAT to identify which was the best system for the current text: our results (Table 2) demonstrate that it is not always the same system which is best. We plan to experi-

ment with this idea in the near future.

Note that the learning or history ideas do not imply training as such. There is no need for a large set of training sentences: the system will learn as it compresses the graph. Even with this type of learning, the system will still be "plug-and-play".

On a more technical level, DEMOCRAT still has some issues that should be looked into. For example, sometimes there are multiple alignments with the same edit cost. At the moment, the system selects one according to an inherent bias (which is present in most edit-distance algorithm implementations). Perhaps taking all possible alignments into account will better capture the words that are used in a similar way.

Also, the graph-walking phase is a greedy search, which does not necessarily find the global best path according to total edge count. This is not necessarily a drawback: in most cases it seems that there is no difference with the global case. To get an idea about why this is the case, consider again the example in figure 2 where the function words serve as "anchors". These words will always be selected, no matter what. However, a future extension could incorporate a Viterbi-like search instead of the current search algorithm.

6 Conclusions

We have implemented a simple system to compile consensus translations from the output of free on-line MT systems, and experimented with two language pairs and a variety of text types. Our results show that there is not a single individual best MT system, so for the general user, it could be useful to make use of software which takes multiple outputs and tries to reconcile them. While DEMOCRAT may not always beat the best individual MT system, it will almost always come close, and will almost never do worse than the worst system. Since our system does not require any training material, and can work immediately with output in any language,[4] we are planning to experiment more extensively with different language pairs and text types, and to explore the pros and cons of global DEMOCRAcy.

[4]See footnote 1 regarding non-spacing writing systems.

Acknowledgements

This work was completed while the second author was on study leave at the Centre for Language Technology, Macquarie University: he is grateful to Robert Dale and the group for their hospitality.

We are most grateful to Simon Zwarts for his evaluation programs, as mentioned in footnote 2, as well as his Python scripts which enabled us to gather automatically multiple translations of large texts avoiding length and repeat usage restrictions.

References

Akiba, Y., Watanabe, T., and Sumita, E. (2002). Using language and translation models to select the best among outputs from multiple MT systems. In *COLING 2002: Proceedings of the 19th International Conference on Computational Linguistics (COLING); Taipei, Taiwan*, pages 8–14.

Bangalore, S., Bordel, G., and Riccardi, G. (2001). Computing consensus translation from multiple machine translation systems. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop; Madonna di Campiglio, Italy.*

Bangalore, S., Murdock, V., and Riccardi, G. (2002). Bootstrapping bilingual data using consensus translation for a multilingual instant messaging system. In *COLING 2002: Proceedings of the 19th International Conference on Computational Linguistics (COLING); Taipei, Taiwan*, pages 50–56.

Brown, P. F., Della Pietra, S., Della Pietra, V., and Mercer, R. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–312.

Brown, R. and Frederking, R. (1995). Applying statistical English language modeling to symbolic machine translation. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 95); Leuven, Belgium*, pages 221–239.

Fiscus, J. (1997). A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER).

In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding; Santa Barbara:CA, USA*, pages 238–245.

Frederking, R. and Nirenburg, S. (1994). Three heads are better than one. In *Fourth Conference on Natural Language Processing; Stuttgart, Germany*, pages 95–100.

Frederking, R., Rudnicky, A., and Hogan, C. (1997). Interactive speech translation in the DIPLOMAT project. In *Spoken Language Translation: Proceedings of a Workshop; Madrid, Spain*, pages 61–66.

Jayaraman, S. and Lavie, A. (2005). Multi-engine machine translation guided by explicit word matching. In *European Association for Machine Translation (EAMT) 10th Annual Conference; Budapest, Hungary.*

Larkey, L. and Croft, B. (1996). Combining classifiers in text categorization. In *Proceedings of the 19th Annual International Conference on Research and Development in Information Retrieval (SIGIR); Zurich, Switzerland*, pages 289–297.

Lavie, A., Vogel, S., Levin, L., Peterson, E., Probst, K., Font Llitjós, A., Reynolds, R., Carbonell, J., and Cohen, R. (2003). Experiments with a Hindi-to-English transfer-based MT system under a miserly data scenario. *ACM Transactions on Asian Language Processing*, 2:143–163.

Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSR*, 163(4):845–848. Original in Russian.

Nomoto, T. (2003). Predictive models of performance in multi-engine machine translation. In *MT Summit IX: Proceedings of the Ninth Machine Translation Summit; New Orleans:LA, USA*, pages 269–276.

Nomoto, T. (2004). Multi-engine machine translation with voted language model. In *42th Annual Meeting of the Association for Computational Linguistics; Barcelona, Spain*, pages 494–501.

Nyberg, E. and Mitamura, T. (1997). A real-time MT system for translating broadcast captions. In *MT Summit VI: Machine Translation Past Present Future; San Diego:CA, USA*, pages 51–57.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: A method for automatic evaluation of machine translation. In *40th Annual Meeting of the Association for Computational Linguistics; Philadelphia:PA, USA*, pages 311–318.

Rayner, M. and Carter, D. (1997). Hybrid processing in the Spoken Language Translator. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97); Munich, Germany*, pages 107–110.

Roth, D. and Zelenko, D. (1998). Part of speech tagging using a network of linear separators. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING) and 36th Annual Meeting of the Association of Computational Linguistics (ACL); Montreal, Canada*, pages 1136–1142.

Turian, J. P., Shen, L., and Melamed, I. D. (2003). Evaluation of machine translation and its evaluation. In *MT Summit IX: Proceedings of the Ninth Machine Translation Summit; New Orleans:LA, USA*, pages 23–28.

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

Appendix

URLs for the source texts and reference translations:

mar: `http://www.marseille-tourisme.com`
euFE and euEG: `http://people.csail.mit.edu/koehn/publications/europarl/`
jv: `http://www.gutenberg.org/etext/5097` and `http://www.gutenberg.org/etext/2488`

URLs for the online MT systems:

ba: `http://world.altavista.com/babelfish`
sy: `http://www.systranbox.com/systran/`
fr: `http://www.freetranslation.com`
tr: `http://www.translate.ru/eng/`
wo: `http://www.worldlingo.com/en/products_services/worldlingo_translator.html`