MISTRAL: A Statistical Machine Translation Decoder for Speech Recognition Lattices

Alexandre Patry, Philippe Langlais

RALI – DIRO
Université de Montréal
{patryale,felipe}@iro.umontreal.ca

Abstract

This paper presents MISTRAL, an open source statistical machine translation decoder dedicated to spoken language translation. While typical machine translation systems take a written text as input, MISTRAL translates word lattices produced by automatic speech recognition systems. The lattices are translated in two passes using a phrase-based model. Our experiments reveal an improvement in BLEU when translating lattices instead of sentences returned by a speech recognition system.

1. Introduction

Automatic Speech Recognition systems (ASR) are intended to recognize a text that was spoken by a performer and Machine Translation systems (MT) are intended to translate written texts from one language to the other. It thus seems natural to put those systems together when translating spoken languages.

It is unfortunately not always that simple because written and spoken texts are not as similar as we may think. Spoken language transcriptions present many problems to a traditional MT system. First, it contains many disfluencies (repetitions, hesitations and other artifacts). Second, it is not easy to recover the structural elements of the message such as punctuation marks, sentence and paragraph boundaries or capitalization. The system presented in this work aims at solving a third problem, namely the robustness of the MT system regarding recognition errors made by the ASR system.

Most speech recognition systems do not convert an audio signal to text greedily one word at a time. They rather generate a lattice where edges correspond to words and nodes to boundaries between words (see Figure 1). ASR systems return the sentence associated with the best path between the source and the sink nodes, which correspond to the beginning and ending of the audio signal. A recognition error happens when a spoken word is unknown to the system or when the wrong path is selected.

In this work, we present MISTRAL, a statistical MT system designed to translate lattices produced by ASR systems. The motivation for lattice translation is the hope that the knowledge embedded in the MT system will help the ASR system to reduce its recognition errors (Ney, 1999).

Word lattices have already been translated by systems based on finite state transducers (Saleem et al., 2004; Matusov et al., 2005; Zhang et al., 2005; Mathias and Byrne, 2006). Our decoder manipulates the same kind of lattices as finite state transducers, but it allows a finer control on pruning policies and it eases the recovery of aggregated informations like individual feature function values.

Translating only the sentence returned by the ASR system is not optimal, but translating a lattice requires a dedicated decoder. A compromise is to translate an n-best list of sentences extracted from the lattice and then select the best

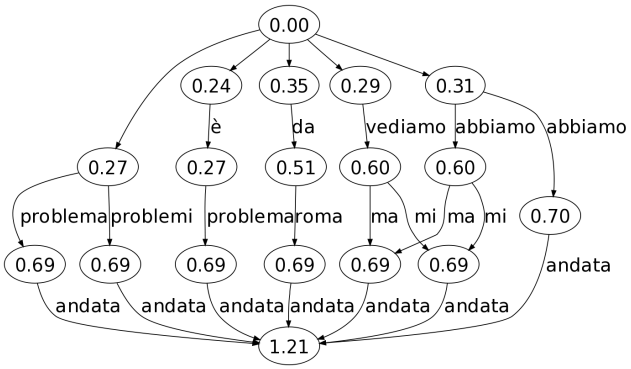

Figure 1: A lattice of words where nodes are labelled with time and edges with words.

translation (Zhang et al., 2004; Quan et al., 2005). Most of the sentences of a given list will be very similar, so translating each of them independently is more computationally expensive than translating a lattice where common parts are factored.

Bertoldi et al. (2007) present an algorithm translating confusion networks. Confusion networks are lattices where each node has at most one predecessor and one successor. Their algorithm deals with non-monotone efficiently translations because word boundaries are the same for all the sentences in a given confusion network. On the other hand, another system must be created and tuned to convert word lattices into confusion networks.

The decoder presented in this work was initially developed for the *International Workshop on Spoken Language Translation* (IWSLT) (Fordyce, 2007; Patry et al., 2007). It is a phrase-based system (Koehn et al., 2003) translating lattices in two passes. The first pass simultaneously recognizes and translates the source sentence and the second pass rescores a list of top-ranked translations obtained from the first pass. MISTRAL is licensed under the *Gnu General Public License*[1] and is available from `http://smtmood.sourceforge.net`.

Other open source decoders translating lattices have been developed. While MISTRAL uses lattices for spoken lan-

[1] `http://www.gnu.org/copyleft/gpl.html`

guage translation, MARIE uses them to encode word reordering (Crego and Mariño, 2007). MOSES (Koehn et al., 2007) translates confusion networks and lattices, but at the time of this writing, its lattice translation algorithm is not documented. We are thus not able to compare our work with it.

This paper is organized as follow. The next section presents the theoretical framework of spoken language translation. We describe our decoder in section 3 and evaluate our complete system in section 4. We finally conclude in section 5.

2. Lattice Translation

A statistical MT system searches the target sentence (t) having the highest probability to translate a given source sentence (s):

$$\mathbf{t}^\star = \underset{\mathbf{t}\in\mathcal{L}_t}{\operatorname{argmax}} \Pr(\mathbf{t}|\mathbf{s}) \tag{1}$$

where $\mathcal{L}_t$ is the set of valid sentences in the target language. When translating a lattice of words (o), both the source and the target sentences are unknown. The statistical MT system thus seeks to solve:

$$\mathbf{t}^\star = \underset{\mathbf{t}\in\mathcal{L}_t}{\operatorname{argmax}} \sum_{\mathbf{p}\in\mathcal{P}_o} \Pr(\mathbf{t}, \mathbf{s_p}, \mathbf{p}|\mathbf{o}) \tag{2}$$

where $\mathcal{P}_o$ is the set of paths from the source node to the sink node in $\mathbf{o}$ and $\mathbf{s_p}$ is the source sentence associated with path $\mathbf{p}$.

The decoder described in this work estimates Eq. (2) under the so-called maximum approximation:

$$\hat{\mathbf{t}} = \underset{\mathbf{t}\in\mathcal{L}_t}{\operatorname{argmax}} \max_{\mathbf{p}\in\mathcal{P}_o} \Pr(\mathbf{t}, \mathbf{s_p}, \mathbf{p}|\mathbf{o}) \tag{3}$$

3. Decoder

A typical phrase-based decoder (Koehn et al., 2003) translates a source sentence one phrase at a time using a translation table (a bilingual dictionary of phrases). A partial target sentence can be extended by the translation of any untranslated phrase in the source sentence. The initial target is made of an empty sentence and a translation is completed when all the source words have been translated.

When translating a lattice, the source sentence is generated along with the target sentence while the lattice is traversed. When a translation is extended with a pair of phrases, the decoder walks the path corresponding to the source phrase in the lattice. The initial translation starts at the source node and a translation is completed when it reaches the sink node.

Conceptually, our decoder combines a source word lattice and a translation table into a translation lattice where nodes correspond to phrase boundaries and edges correspond to pair of phrases (see Figure 2). As the translation lattices are created during traversal, they can be pruned and traversed effectively regardless of their size.

The search for a translation will fail if all the paths in the lattice contain a phrase that is unknown to the translation table. To avoid this kind of failure, unknown words following a partial translation that cannot be extended are all considered to be translated by themselves.

Italian	English
non	not, don't
fumatory	smoker
non fumatory	non-smoking

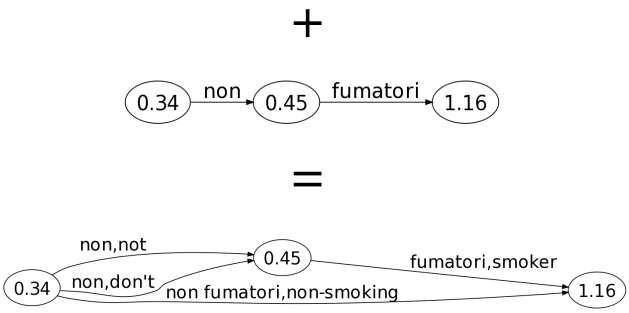

Figure 2: The word lattice is combined with the translation table to generate an implicit translation lattice.

The probability of a translation is approximated using an exponential model. A first model is used to generate an n-best list of translations and a second model to reorder this list.

3.1. Model

To approximate the value of $\Pr(\mathbf{t}, \mathbf{s_p}, \mathbf{p}|\mathbf{o})$ in Eq. (3), we use an exponential model:

$$\begin{aligned}\hat{\mathbf{t}} &= \underset{\mathbf{t}\in\mathcal{L}_t}{\operatorname{argmax}} \max_{\mathbf{p}\in\mathcal{P}_o} Z^{-1} \exp\left(\sum_r \lambda_r f_r(\mathbf{t}, \mathbf{s_p}, \mathbf{p}, \mathbf{o})\right) \\ &= \underset{\mathbf{t}\in\mathcal{L}_t}{\operatorname{argmax}} \max_{\mathbf{p}\in\mathcal{P}_o} \sum_r \lambda_r f_r(\mathbf{t}, \mathbf{s_p}, \mathbf{p}, \mathbf{o})\end{aligned} \tag{4}$$

where Z is a normalization factor, $f_r(\cdot,\cdot,\cdot,\cdot)$ are feature functions returning real values and λ_r are free parameters weighting the feature functions.

As MISTRAL is implemented with a modular framework (Patry et al., 2006), feature functions are easy to add. At the time of this writing, it supports ASR scores in lattices, source and target language models, scores in translation tables and IBM model 1 scores (Och et al., 2004).

3.2. First pass

Exploring the entire translation lattice would be intractable, thus it must be pruned. A common pruning technique in statistical MT consist in grouping together partial translations covering a similar portion of the source sentence and then considering only the top ranked translations of each group.

Partial translations are usually grouped by the number of words they translate. As the number of source words is not fixed when translating a lattice, we propose to group translations by the portion of audio signal they cover. The audio signal is divided in time slices of equal duration, and a group is assigned to each time slice.

The groups are explored in chronological order. When a group is considered, it is pruned and the remaining translations are moved to a temporary group. Once all the translations of the temporary group are extended, we check if the

considered group contains new translations, meaning that at least one partial translation was in the same time slice before and after it was extended. If it is the case, the group is explored again, otherwise the next group is considered.
A group is pruned in two steps. The so-called histogram pruning, which keeps only a fix number of translations, is applied in the former step. The remaining translations are checked for recombination in the latter step. Two or more translations can be recombined if their future path and scores are identical. This happens when they correspond to the same node in the lattice, their last source words are the same and their last target words are the same (because of the language models).

3.3. Second pass

Some feature functions require more computation than others. The IBM model 1 feature function considers all the pairs of source and target words of a translation. It thus makes recombination possible only for identical translations pointing at the same node in the lattice. Other feature functions limiting recombinations are language models, because they require the recombined translations to be suffixed by the same words.
In order to use these features without increasing the decoding time too much, we consider them only in the second pass, which rescores a list of translations produced in the first pass.

4. Experiments

In this section, we evaluate MISTRAL on a wide range of configurations and compare the best configuration against a fair baseline.

4.1. Data

We evaluated MISTRAL on the corpus that was provided for the IWSLT 2007 Italian-English shared task. This corpus is composed of Italian spontaneous conversations in the travel domain and their English translations. It is divided in train (IN-DOMAIN) and development (DEV) sections containing respectively 19,722 and 996 sentence pairs. Lattices are provided only for the DEV corpus. These lattices are scored with an acoustic model and a language model.
Because the IN-DOMAIN corpus is small, we also trained our models on the Italian-English section of the proceedings of the European Parliament (EUROPARL), which contains more than 928,000 sentence pairs (Koehn, 2005). Before training, we lowercased all the material and we removed the punctuation marks, since our lattices do not contain any.
We trained one translation table on IN-DOMAIN and another on EUROPARL using the *grow-diag-final* heuristics (Koehn et al., 2003), which extracts phrase pairs from a word alignment that was produced by GIZA++ (Och and Ney, 2000). Knowing that the corpora contain many dates and numbers, we manually created a third translation table made of translations for days, months and numbers.
For all the experiments, the first pass is tuned on the first 300 sentences of DEV, the second pass on the following 300 sentences and the evaluation metrics are computed on the remaining 396 sentences.

4.2. Evaluation

One specificity of lattice translation compared to typical MT is the generation of the source sentence along with the target sentence. We compare the source sentences extracted from the lattices with the original transcriptions using word-error-rate (WER) and the target with the reference translations using BLEU (Papineni et al., 2001). Both scores take a value between zero and one, but reported figures are multiplied by 100 to enhance readability.

4.3. Feature Functions

The following feature functions are used in the first pass:

- ASR scores.
- Two English and two Italian trigrams trained on the IN-DOMAIN and EUROPARL corpora.
- The number of words in the source and in the target sentences.
- The number of phrases in the translation.
- The translation probability of the phrases in both translation directions estimated by relative frequencies.
- The lexical weighting of the phrases in both translation directions. Lexical weighting estimates the probability of a phrase at the word alignment level (Koehn et al., 2003).
- Three binary functions associating a pair of phrases with its origin (IN-DOMAIN, EUROPARL or manually created).

and the following feature functions are added for the second pass:

- Two English and two Italian 4-grams trained on the IN-DOMAIN and EUROPARL corpora.
- Four IBM model 1 translation scores from models trained on IN-DOMAIN and EUROPARL corpora in both translation directions.

The weights of the different feature functions are optimized on BLEU using the downhill simplex algorithm (Press et al., 1992). All the weights are initialized to 0.1 except the weights of ASR features, which get a higher values in order to start with good source sentences.
Running the decoder whenever weights are updated is computationally expensive. The weights are thus optimized on n-best lists as suggested in Och et al. (2004). A first set of n-best lists are generated from the initial configuration. When the weights are updated by the optimization algorithm, those lists are reordered according to the new weights. Once optimal weights are found, a new set of n-best lists is generated. The optimization algorithm iterates as long as the new lists are different from the previous ones. First pass models needed less than 10 iterations to converge. As the list of translations is fixed for the second pass, only one iteration is needed. An implementation of this tuning algorithm is packaged with MISTRAL.

Lattice	1st Pass		2nd Pass	
	WER	BLEU	WER	BLEU
ASR scores	49.33	10.87	49.36	10.99
posterior	12.58	16.51	12.78	18.20
posterior, pruned	12.24	19.00	12.24	19.76

Table 1: Evaluation of translations produced using ASR scores and posterior probability on full and pruned lattices.

4.4. Word Lattices

As the lattices in DEV are scored with an acoustic model and a language model, those are the ASR scores that we consider at first. The results, which are presented in Table 1, are quite deceptive with a WER of 49.36 and a BLEU of 10.99. A closer look at the data reveals that the acoustic score fluctuates a lot and sort of shadows all the other scores.

In order to reduce the variance of ASR scores, we augmented each edge with its posterior probability using the *lattice-tool* utility packaged in the SRILM toolkit (Stolcke, 2002). The posterior probability of an edge is computed by summing the scores of all the paths containing the edge normalized over the sum of the scores of all the paths in the lattice (Wessel et al., 2000). When posterior probability is the only ASR score, WER decreases to 12.78 and BLEU increases to 18.20.

Word lattices typically encode many low-probability paths. Because translations of bad source sentences are of no interest, we investigated whether pruning the lattice is a good strategy. We did a third experiment where we pruned the edges having a posterior probability smaller than one percent of the posterior probability of the best edge starting from the same node. Translating pruned lattices improves BLEU (see Table 1) and decreases the average number of edges per spoken word from 360 to 2.7. It also reduces the decoding time by seven.

In all the experiments of this section, only the 10 best translations of groups spanning 0.1 second are considered and the second pass rescores 2000 translations.

As pruned lattices scored with posterior probabilities yield the best BLEU and the best WER, results presented in further sections are obtained from those lattices.

4.5. Pruning

MISTRAL groups together partial translations that cover similar portion of audio signal and only considers the top-ranked candidates of each group (see section 3.2). In this section, we vary the duration of time slices and the number of translations extended per second. For example, when groups duration is 0.1 second and 10 translations are considered for each group, we say that 100 translations are extended per second. The evaluation of the first pass is presented in Figure 3 and the evaluation of the second pass in Figure 4.

Since many figures are very close, it is somehow delicate to draw strong conclusions. We can nonetheless observe general trends in the results. One trend we observe is that the second pass generally improves BLEU without changing too much WER, which vary between 11.89 and 12.50 for all the experiments. We also observe that the four best BLEU scores are obtained when the groups are conditioned on short time intervals (0.1 or 0.5 second).

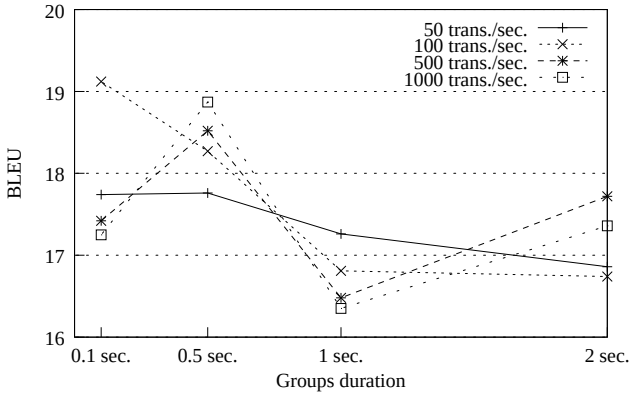

Figure 3: Comparison of different pruning configurations on the first pass.

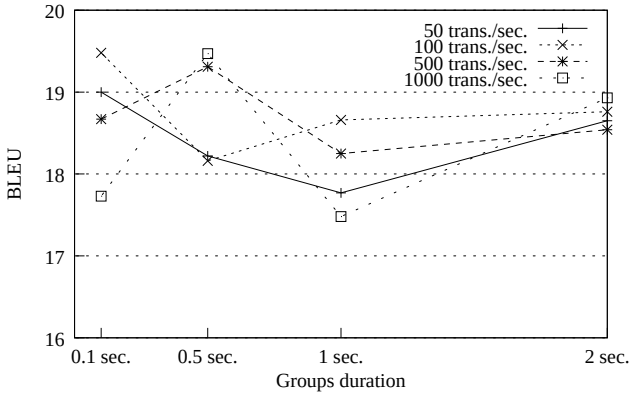

Figure 4: Comparison of different pruning configurations on the second pass.

An unexpected result is that extending more translations can worsen the objective measure. This might be explained by the tuning algorithm that got stuck in a bad local optimum.

In the sequel, groups are conditioned on time intervals lasting 0.1 second and only the 10 best translations of each group are extended.

4.6. N-Best Lists

Both the tuning algorithm and the second pass take an n-best list as input. Figure 5 presents the evaluation of MISTRAL when the size of the n-best lists varies.

While 100 translations do not seem to be enough for tuning and rescoring, 500 and more yield comparable figures. As it obtains the best BLEU, we set the size of n-best lists to 2000 for the comparison with the baselines in the following section.

4.7. Lattice vs ASR Output

In this section we compare translations from pruned lattices with translations from automatic transcription generated by

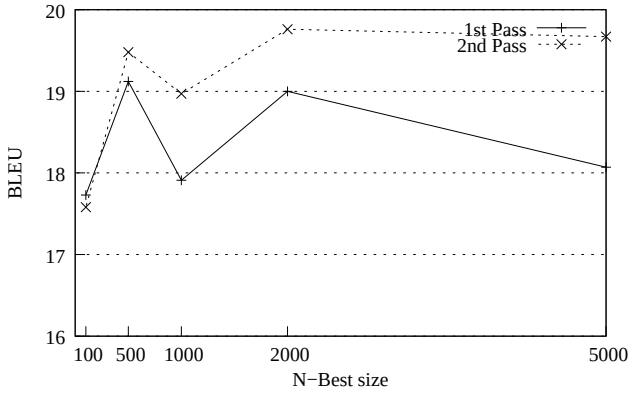

Figure 5: Comparison of n-best lists of different sizes.

Source	1st Pass		2nd Pass	
	WER	BLEU	WER	BLEU
Reference	0	19.28	0	20.79
ASR	11.90	17.39	11.90	19.05
Lattice	12.04	19.00	12.24	19.76

Table 2: Comparison of MISTRAL when it translates the reference transcription, the sentences returned by the ASR system and the lattices.

the ASR system and translations from the reference transcription provided by the organizers of the IWSLT shared task. Results are presented in Table 2.

The ASR sentences are obtained using the viterbi algorithm on the lattices with a weight of 10 for the language model and weights of one for the acoustic model and the word penalty. Because MISTRAL takes lattices as input, ASR output and reference transcriptions are converted into lattices where each word has an arbitrary duration of one second.

All configurations are tuned and rescored with n-best lists of size 2000. Lattices are translated with 10 groups per second and only the 10-best translations of each group are considered. Translations of reference transcriptions and ASR output are grouped by their number of translated words and only the 100-best translations of each group are considered.

Our starting hypothesis is that translating lattices should improve translation quality because the translation model should help the ASR system to reduce its recognition errors. Indeed, translation quality improved according to BLEU, but this is not corroborated with a decreasing WER. A qualitative evaluation reveals that many differences between the ASR output and the sentences recognized by MISTRAL are due to insertion or deletion of small words that do not convey much information (like *sì*, *il* or *è*). We thus think the BLEU improvement is in part explained by the added flexibility of lattice translation, which can distort a little bit the source sentences to make them easier to translate.

To validate our implementation, we translated the ASR output and the reference transcriptions with MOSES (Koehn et al., 2007). When given the same parameters, both MISTRAL and MOSES produce exactly the same translations.

5. Conclusion

We presented and evaluated MISTRAL, a decoder for spoken language translation. To the best of our knowledge, MISTRAL, which is available at `http://smtmood.sourceforge.net`, is the first open source decoder dedicated to the translation of lattices produced by an ASR system.

In our experiments, translating pruned lattices yields better BLEU scores than translating the best sentences returned by an ASR system. To our surprise, this improvement was not corroborated by an improvement in WER as we initially tough. While those results are encouraging, they still need to be confirmed on a larger dataset and on other language pairs.

MISTRAL is a good baseline system for spoken language translation and a good starting point to create more sophisticated lattice decoders. Our future work will be oriented toward the addition of new feature functions, the handling of non-monotone translations and the translation of arbitrary lattices.

6. References

Nicola Bertoldi, Richard Zens, and Marcello Federico. 2007. Speech translation by confusion network decoding. In *Proceedings of the 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, USA, April.

Josep M. Crego and José B. Mariño. 2007. Extending MARIE: an N-gram-based SMT decoder. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 213–216, Prague, Czech Republic, June. Association for Computational Linguistics.

Cameron Shaw Fordyce. 2007. Overview of the iwslt 2007 evaluation campaign. In *Proceedings of IWSLT 2007*, Trento, Italy, October.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *2nd Workshop on EBMT of MT-Summit X*.

L. Mathias and W. Byrne. 2006. Statistical phrase-based

speech translation. In *IEEE Conference on Acoustics, Speech and Signal Processing*.

E. Matusov, S. Kanthak, and H. Ney. 2005. On the integration of speech recognition and statistical machine translation. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech)*, September.

Hermann Ney. 1999. Speech translation: coupling of recognition and translation. In *Proceedings of ICASSP*.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Conference of the Association for Computational Linguistic (ACL)*, pages 440–447, Hong Kong, China, October.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 161–168, Boston, Massachusetts, USA, May. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Alexandre Patry, Fabrizo Gotti, and Philippe Langlais. 2006. MOOD: A modular object-oriented decoder for statistical machine translation. In *5th LREC*, pages 709–714, Genoa, Italy, May.

Alexandre Patry, Philippe Langlais, and Frédéric Béchet. 2007. MISTRAL: A lattice translation system for IWSLT 2007. In *Proceedings of IWSLT 2007*, Trento, Italy, October.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 1992. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA.

H.V. Quan, M. Federico, and Cettolo M. 2005. Integrated n-best re-ranking for spoken language translation. In *Interspeech 2005 - Eurospeech - 9th European Conference on Speech Communication and Technology*.

S. Saleem, S.-C. Jou, S. Vogel, and T. Schultz. 2004. Using word lattice information for a tighter coupling in speech translation systems. In *Proc. ICSLP*, Jeju Island, Korea, Oct.

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

Frank Wessel, Ralf Schlüter, and Hermann Ney. 2000. Using posterior word probabilities for improved speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1587–1590, Istanbul, Turkey, June.

Ruiqiang Zhang, Genichiro Kikui, Hirofumi Yamamoto, Taro Watanabe, Frank Soong, and Wai Kit Lo. 2004. A unified approach in speech-to-speech translation: integrating features of speech recognition and machine translation. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1168, Morristown, NJ, USA. Association for Computational Linguistics.

Ruiqian Zhang, Genichiro Kikui, Hirofumi Yamamoto, and Wai-Kit Lo. 2005. A decoding algorithm for word lattice translation in speech translation. In *Proceedings of 2005 International Workshop on Spoken Language Translation*.