MISTRAL: A Lattice Translation System for IWSLT 2007

Alexandre Patry[1] Philippe Langlais[1] Frédéric Béchet[2]

[1]Université de Montréal

[2]University of Avignon

International Workshop on Spoken Language Translation, 2007

Overview of Mistral

The main characteristics of MISTRAL are:

- it uses a phrase-based model
- it works directly on lattices
- it scores (and rescores) hypotheses with a log-linear model
- it uses a beam search algorithm to organise the search space

General algorithm

MISTRAL uses the following algorithm to translate a lattice:

1. Push ⟨*empty source, empty target, lattice's start node*⟩ on the stack.
2. Extend and prune incomplete hypotheses.
3. Return the best hypothesis that points at the lattice's end node.

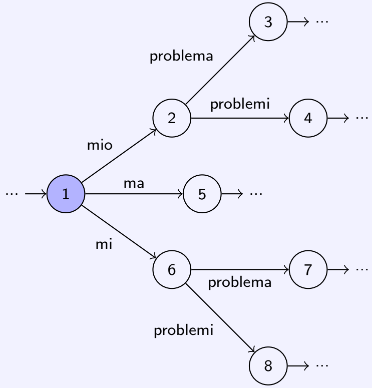
General algorithm

MISTRAL uses the following algorithm to translate a lattice:

1. Push ⟨*empty source, empty target, lattice's start node*⟩ on the stack.
2. Extend and prune incomplete hypotheses.
3. Return the best hypothesis that points at the lattice's end node.

Example of hypothesis expansion

⟨*è is, it is, node 1*⟩



Italian	English
mio	my
mio problema	my problem
mio problema	my concern
mi problemi	my problems

● in the stack

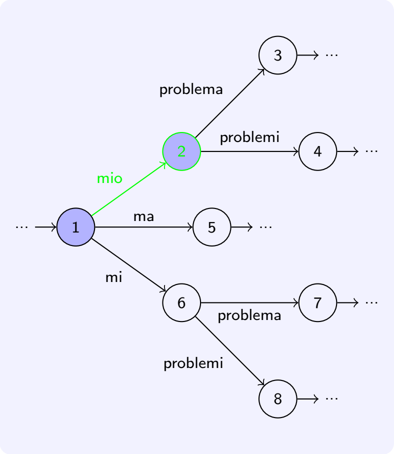
○ not explored yet

◌ explored

◌ pruned

Example of hypothesis expansion

⟨ è is mio, it is my, node 2 ⟩



Italian	English
mio	my
mio problema	my problem
mio problema	my concern
mi problemi	my problems

● in the stack

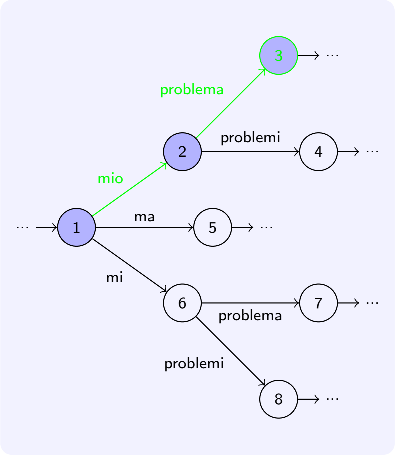
○ not explored yet

◌ explored

◌ pruned

Example of hypothesis expansion

⟨ *è is mio problema*, *it is my problem*, *node 3* ⟩

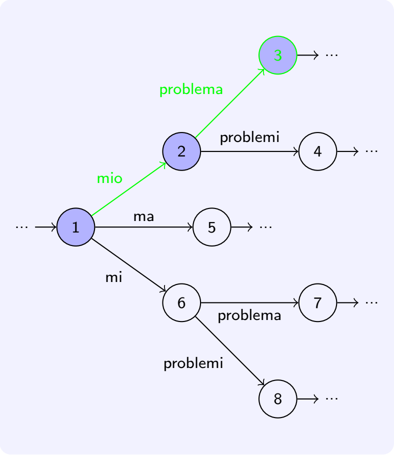


Italian	English
mio	my
mio problema	my problem
mio problema	my concern
mi problemi	my problems

- ● in the stack
- ○ not explored yet
- ◌ explored
- ⬚ pruned

Example of hypothesis expansion

⟨ *è is mio problema, it is my concern, node 3* ⟩



Italian	English
mio	my
mio problema	my problem
mio problema	my concern
mi problemi	my problems

● in the stack

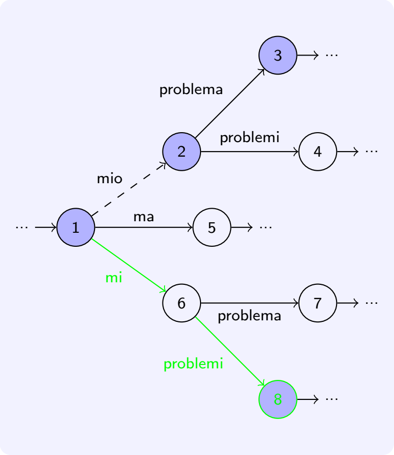
○ not explored yet

◌ explored

◌ pruned

Example of hypothesis expansion

⟨ *è is mi problemi, it is my problems, node 8* ⟩



Italian	English
mio	my
mio problema	my problem
mio problema	my concern
mi problemi	my problems

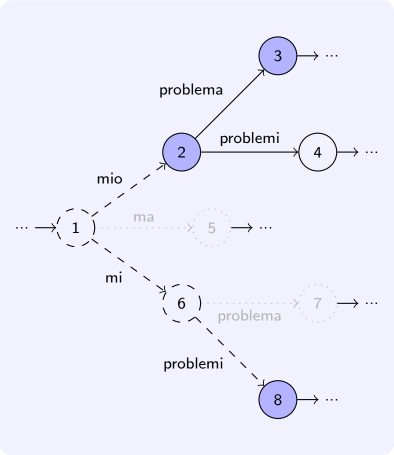
● in the stack

○ not explored yet

◌ explored

◌ pruned

Example of hypothesis expansion



Italian	English
mio	my
mio problema	my problem
mio problema	my concern
mi problemi	my problems

● in the stack

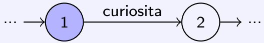
○ not explored yet

◌ explored

◌ pruned

Unknown words when no expansion is possible

$$\langle \ldots \mathit{ecco\ per}, \ldots \mathit{here's\ for}, \mathit{node\ 1} \rangle$$



Italian	English
EMPTY	

● in the stack

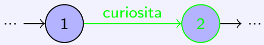
○ not explored yet

◌ explored

⬚ pruned

Unknown words when no expansion is possible

⟨ *. . . ecco per curiosità, . . . here's for curiosità, node 2* ⟩



Italian	English
EMPTY	

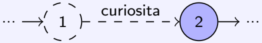
● in the stack

○ not explored yet

◌ explored

◌ pruned

Unknown words when no expansion is possible



Italian	English
EMPTY	

● in the stack

○ not explored yet

◌ explored

⬚ pruned

Beam search

The search space is organised with a beam search:

- One stack for each time slice of 0.1 second.
- Breadth first search for each stack (it can happen when a word is shorter than 0.1 second). When this happens, pruning is done before the exploration of each depth.

Pruning of a stack

The pruning of a stack is done in two steps:

1. Keep the 50 best hypotheses.
2. Recombine the remaining hypotheses sharing:
    - the same node
    - their last two source words (source lm)
    - their last two target words (target lm)

Exponential model

Each hypothesis is scored (and rescored) with an exponential model:

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \max_{\mathbf{f}} \sum_{r=1}^{R} \lambda_r h_r(\mathbf{e}, \mathbf{f}, \mathbf{o})$$

where **f** and **e** are the source and target sentences and **o** is the lattice returned the ASR system.

Evaluation protocol

We evaluated MISTRAL on the Italian-English track using the following protocol:

1. Train translation tables on the training corpus and EUROPARL.
2. Tune the first pass on the first 300 sentences of the DEV corpus.
3. Tune the rescoring pass on the 300 following sentences of the DEV corpus.
4. Test on the remaining 396 sentences of the DEV corpus.

Training

We created one language model and one translation table for each of the following corpora:

- IWSLT training data (19 722 sentence pairs)
- EUROPARL corpus ($>$ 928,000 sentence pairs)

We manually created a third translation table containing 122 rules for days, months and numbers.

Our final translation table is the concatenation of those three.

First pass

The following feature functions were used for the first pass:

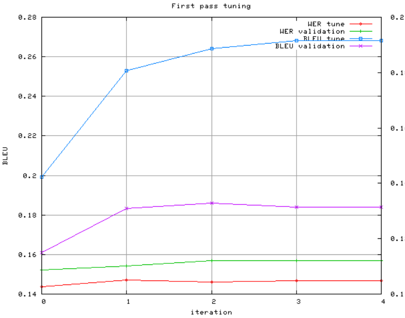
- Posterior probability of the path in the lattice.
- Two source and two target trigrams.
- Source and target word penalties.
- Translation table scores:
  - relative frequencies
  - lexical probabilities
  - constant penalty
  - three binary features associating an entry with its corpus

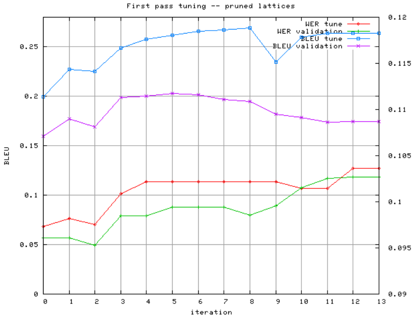
First pass tuning

The first pass weights are tuned as follow:

1. Initialise the weight of the posterior probability to 10 and the other weights to 0.1.
2. Extract the 500 best translations from each lattice of the first 200 sentences.
3. Optimize BLEU on those N-Best lists using the downhill simplex algorithm.
4. If the weights were updated, go to 2.
5. Use the last 100 sentences as a validation corpus to select the weights of the best iteration.

First pass tuning





Rescoring

The following feature functions are added for the rescoring pass:

- Two source and two target 4-grams.
- Lexical probabilities of the complete sentences in both translation directions.

Rescoring tuning

The rescoring weights are tuned on the next 300 sentences of the DEV corpus as follow:

1. Initialise the weights of the new feature functions to 0.1.
2. Run the first pass to extract the 500 best translations of each lattice.
3. Optimize BLEU on those N-Best lists using the downhill simplex algorithm.

Results

System	1st Pass		Rescoring	
	WER	BLEU	WER	BLEU
Ref	0	20.09	0	21.27
1-best	11.90	17.97	11.90	19.37
Opt. on BLEU	12.04	17.08	12.07	19.24
Opt. on WER and BLEU	11.81	17.58	11.87	18.93
Opt. on BLEU, pruned lattices	10.96	19.21	11.04	20.28

We used MISTRAL on the reference and the 1-best to have an idea of the performance we should expect.

Results

System	1st Pass		Rescoring	
	WER	BLEU	WER	BLEU
Ref	0	20.09	0	21.27
1-best	11.90	17.97	11.90	19.37
Opt. on BLEU	12.04	17.08	12.07	19.24
Opt. on WER and BLEU	11.81	17.58	11.87	18.93
Opt. on BLEU, pruned lattices	10.96	19.21	11.04	20.28

Disappointing results when our system is run on unpruned lattices. Worse WER and BLEU than 1-best.

Results

System	1st Pass		Rescoring	
	WER	BLEU	WER	BLEU
Ref	0	20.09	0	21.27
1-best	11.90	17.97	11.90	19.37
Opt. on BLEU	12.04	17.08	12.07	19.24
Opt. on WER and BLEU	11.81	17.58	11.87	18.93
Opt. on BLEU, pruned lattices	10.96	19.21	11.04	20.28

Optimizing on the harmonic mean of WER and BLEU diminishes WER at the expense of BLEU.

Results

System	1st Pass		Rescoring	
	WER	BLEU	WER	BLEU
Ref	0	20.09	0	21.27
1-best	11.90	17.97	11.90	19.37
Opt. on BLEU	12.04	17.08	12.07	19.24
Opt. on WER and BLEU	11.81	17.58	11.87	18.93
Opt. on BLEU, pruned lattices	10.96	19.21	11.04	20.28

Our best results were obtained when we optimized on BLEU and pruned the the lattices. An edge is pruned if its post-probability is lower than 1% of the highest post-probability of all edges starting at the same node.

Results

System	1st Pass		Rescoring	
	WER	BLEU	WER	BLEU
Ref	0	20.09	0	21.27
1-best	11.90	17.97	11.90	19.37
Opt. on BLEU	12.04	17.08	12.07	19.24
Opt. on WER and BLEU	11.81	17.58	11.87	18.93
Opt. on BLEU, pruned lattices	10.96	19.21	11.04	20.28

The average number of word hypotheses per spoken word passes from 360 to 2.7 after pruning.

Results

System	1st Pass		Rescoring	
	WER	BLEU	WER	BLEU
Ref	0	20.09	0	21.27
1-best	11.90	17.97	11.90	19.37
Opt. on BLEU	12.04	17.08	12.07	19.24
Opt. on WER and BLEU	11.81	17.58	11.87	18.93
Opt. on BLEU, pruned lattices	10.96	19.21	11.04	20.28

Even translating the reference yielded poor results. Is it our model or our implementation that is at fault?

Comparison with MOSES

Input	System	1st Pass	Rescoring
		BLEU	
Ref	MISTRAL	20.09	21.27
	MOSES w/o distortion	20.91	-
1-best	MISTRAL	17.97	19.37
	MOSES w/o distortion	18.99	-

MOSES was systematically better on the 1st pass, but its BLEU scores are low as well. The models we trained are probably at fault.

Comparison with MOSES

Input	System	1st Pass	Rescoring
		BLEU	
Ref	MISTRAL	20.09	21.27
	MOSES w/o distortion	20.91	-
1-best	MISTRAL	17.97	19.37
	MOSES w/o distortion	18.99	-

Later experiments showed us that results of MISTRAL are similar to those of MOSES when the translation table is not pruned. It is because we did not consider the word penalty and the language models but only the translation table scores during pruning.

A note on features

We made the following observations about the features and their weights:

- The features that had the highest weights were the one related to ASR (post-probability, italian trigrams and penalties).
- The EUROPARL translation table helped us gain more than 1 point in BLEU.
- Same observation for the binary feature functions associating an entry of the translation table to its origin.
- When rescoring, 4-grams did not help, lexical probabilities alone did the job.

Post-processing

The capitalisation was restored with the DISAMBIG tool from the SRILM toolkit. Each word was ambiguously capitalized or not.

Only final punctuation marks were restored by a Naïve Bayes classifier taking as input the first word of each sentence.

Both models were trained on the training corpus supplied for the shared task.

Shared task results

System	BLEU			
	Before	C	P	C + P
Official run	21.03	18.66	16.12	13.90
Updated system	23.81	20.75	17.60	16.17

Bugs in MISTRAL were fixed since we submitted our official results, so we repeated the shared task with our updated system.

Future works

MISTRAL is a young system and many things were overlooked due to a lack of time:

- The pruning parameters were not thoroughly examined (stack size, nbest list sizes, duration of a time slice).
- We have always started tuning from the same point.
- No statistical significance test were run.
- We should test our system on a bigger corpus.

Conclusion

We presented MISTRAL, a phrase-based decoder working directly on lattices.

Our results are disappointing in two ways:

- BLEU scores are low in general
- it does not surpass clearly the 1-best baseline