Automatic Line Segmentation and Ground-Truth Alignment of Handwritten Documents

Théodore Bluche, Bastien Moysset, Christopher Kermorvant

Introduction

To train recognition systems, we need **annotated (position and transcript) lines of text**

On the web, we can retrieve many **transcribed images without line positions** ⟶ we have to **map the transcript to the image**

In the litterature, line positions are assumed to be known or reliably obtained with automatic methods

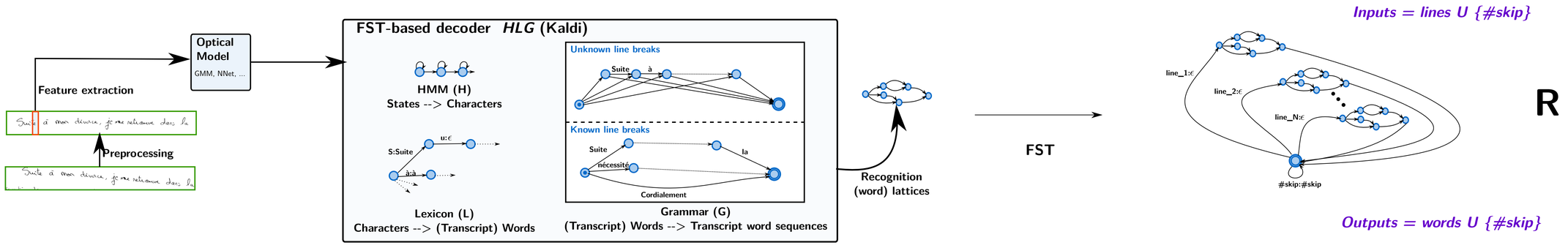
We propose a method able to ...

- ☆ consider **several segmentation hypotheses**
- ☆ **jointly find** the segmentation and transcript mapping
- ☆ **reject lines** in the segmentation, which content is not in the transcript
- ☆ perform the **mapping with a recognition system**, constrained by the transcript

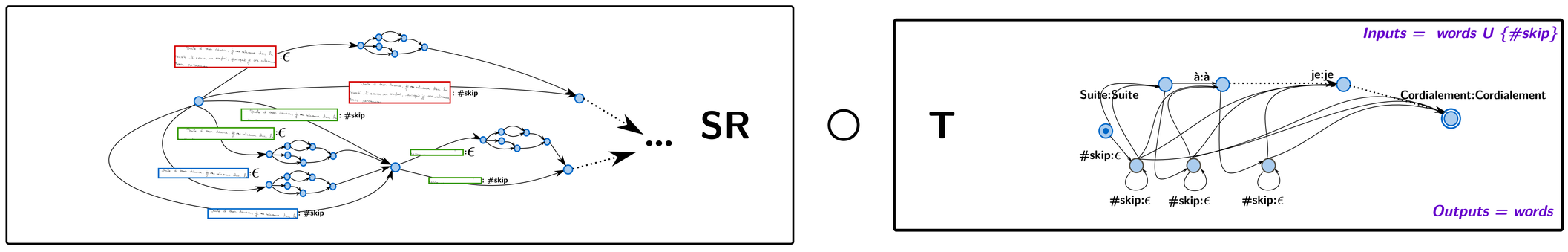
Method

Segmentation Hypotheses Transducer

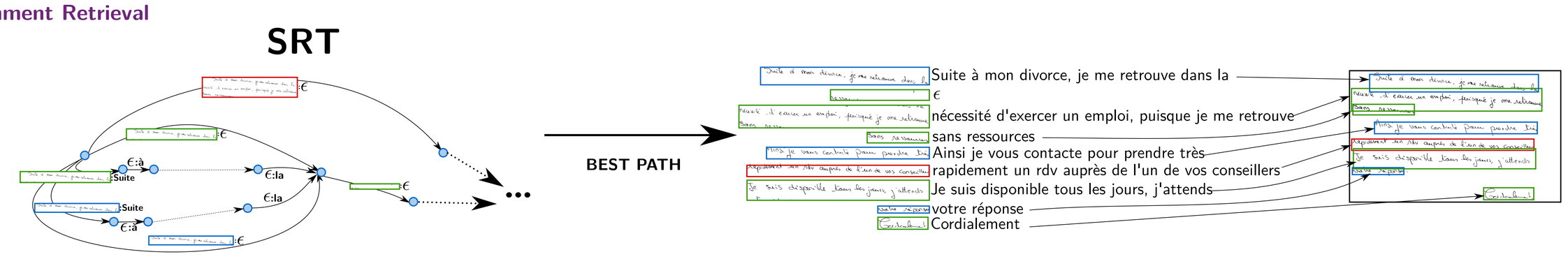
Recognition Hypotheses Transducer



Transcript Transducer



Alignment Retrieval



Results

Analysis

- ☆ To evaluate the method, we have to measure the quality of the **segmentation** and of the **mapping**
- ☆ We applied the method on public databases for which we know the line positions and transcript (Rimes, IAM)
- ☆ **Segmentation error = ZoneMap**

  ZoneMap aligns bounding boxes from a reference and an hypothesis in terms of *Matches, Misses, Merges, Splits and False Alarms*

  The error counts *black pixels that are missed or falsely included* in an hypothesis segmentation w.r.t the reference segmentation
- ☆ **Mapping error = Edit Distance**

  We use the bounding box matching found with ZoneMap

  For each configuration, we count the number of word subsitutions, deletions, and insertions

  (Note: misses -> deletions, false alarms -> insertions)

We evaluated the **different aspects** of the method

- ☆ Influence of mapping on segmentation quality
- ☆ Benefits of keeping multiple segmentation hypotheses
- ☆ Influence of the different constraints and benefits of knowing line breaks in the transcript
- ☆ Influence of the recognition system

	Seg.Err.	Map.Err.
Shredding		
Segmentation only	1.56	-
Segmentation + Mapping	0.77	1.24
Rectangle Filtering		
Segmentation only	4.90	-
Segmentation + Mapping	6.03	4.48
Projection profile		
Segmentation only	1.56	-
Segmentation + Mapping	0.87	0.97
All three segmentations		
Segmentation only	282.38	-
Segmentation + Mapping	0.90	1.22
No transcript constraint (SR only)	0.75	3.28
No recognition order (no G in reco)	88.85	90.24
Known line break symbols	0.82	0.22
Optical Model		
GMM	0.90	1.22
BLSTM-RNN	0.80	0.11
BLSTM-RNN (Rimes)	1.06	0.16

Results on IAM (dev)

A practical usage: creation of training material

For the **Maurdor competition**, we had :

- ☆ Annotated zones of text (either 1 or more lines)
- ☆ But **no line position** for multi-line zones
- ☆ However, the transcript contains line break symbols

Method

1 - Train an RNN on single line zones
2 - Use it to map the transcript of multi-line zones
3 - Train a new RNN with the new material and go back to **2**

RNN Training Material	# lines / % of max	WER
Single-line zones	7,310 / 63.0	54.7%
AutoSegMap (iteration 1)	10,570 / 91.1	43.8%
AutoSegMap (iteration 2)	10,925 / 94.1	**35.2%**

Limitations - Future Work

- ☆ The current segmentation FST can only handle simple layouts
  --> we need to be **able to cope with multi-columns, side notes, etc.** with a more elaborated graph
- ☆ The segmentation FST could be improved if the segmentation algorithm returned positions with **confidence scores**
- ☆ The recognition is very constrained, and allows to only recognize transcript words
  --> an **implementation of line rejection at this level** could be beneficial
- ☆ The method cannot cope with transcript errors, as in other publications
  --> it could be implemented in the FST

Conclusions

We implemented several trivial constraints derived from the knowledge of the transcript.

- ☆ the transcript order in the decoding graph enables a **quick recognition and is crucial for a good mapping** even with a recognition system which has not been adapted
- ☆ the transcript FST is important for a **mapping that is consistent at the document level** (i.e. the same part of the transcript is not mapped to several lines)
- ☆ finding a good mapping with this method **generally improves the segmentation** (less lines are falsely accepted, but some are wrongly discarded)
- ☆ keeping several segmentation hypotheses is not always better than the best segmentation, but **good since we do not know *a priori* which segmentation algorithm will be better**

We applied this method to retrieve more training material for recognition systems

- ☆ in the Maurdor evaluation, this accounted for a **35.6% relative improvement** and was crucial for winning the competition
- ☆ in other projects, this helped to quickly create annotated databases for handwriting recognition system training

This work was partially funded by the French Defense Agency (DGA) through the Maurdor research contract with Airbus Defence and Space (Cassidian), and by the French Grand Emprunt-Investissements d'Avenir program through the PACTE project.

{tb,bm,ck}@a2ia.com