# Highlighting Matched and Mismatched Segments in Translation Memory Output through Sub-Tree Alignment

**Ventsislav Zhechev**

*Translating and the Computer 32 Conference, London, UK* *19 November 2010*

# Outline

- Translation Memory Backend
- Sub-Tree Alignment
- Translation-Alignment Algorithm
- Evaluating the usefulness of Statistical Machine Translation
- Future Work
- Conclusion

# Translation Memory Backend

- A PostgreSQL database containing the plain TM data
- Can perform fuzzy matching based on a fast character-based Levenstein-distance search
- The Levenstein-based distance of the fuzzy match is normalised by the number of characters in the shorter sentence
- Integrate with a proper TM in the future

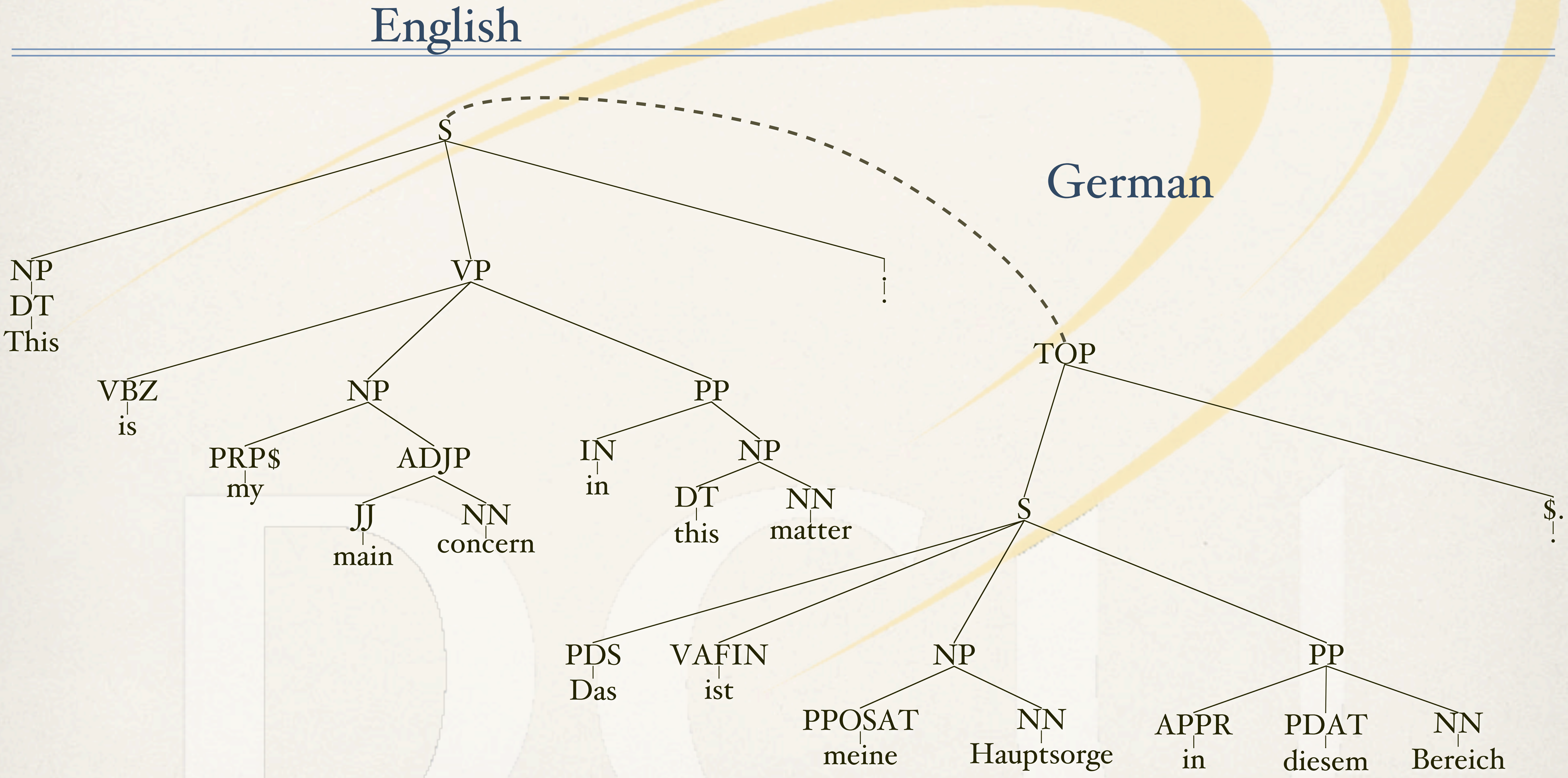
# Sub-Tree Alignment

- Main use: for generating training resources for Syntax-Based Machine Translation
  - i.e. Parallel Treebanks

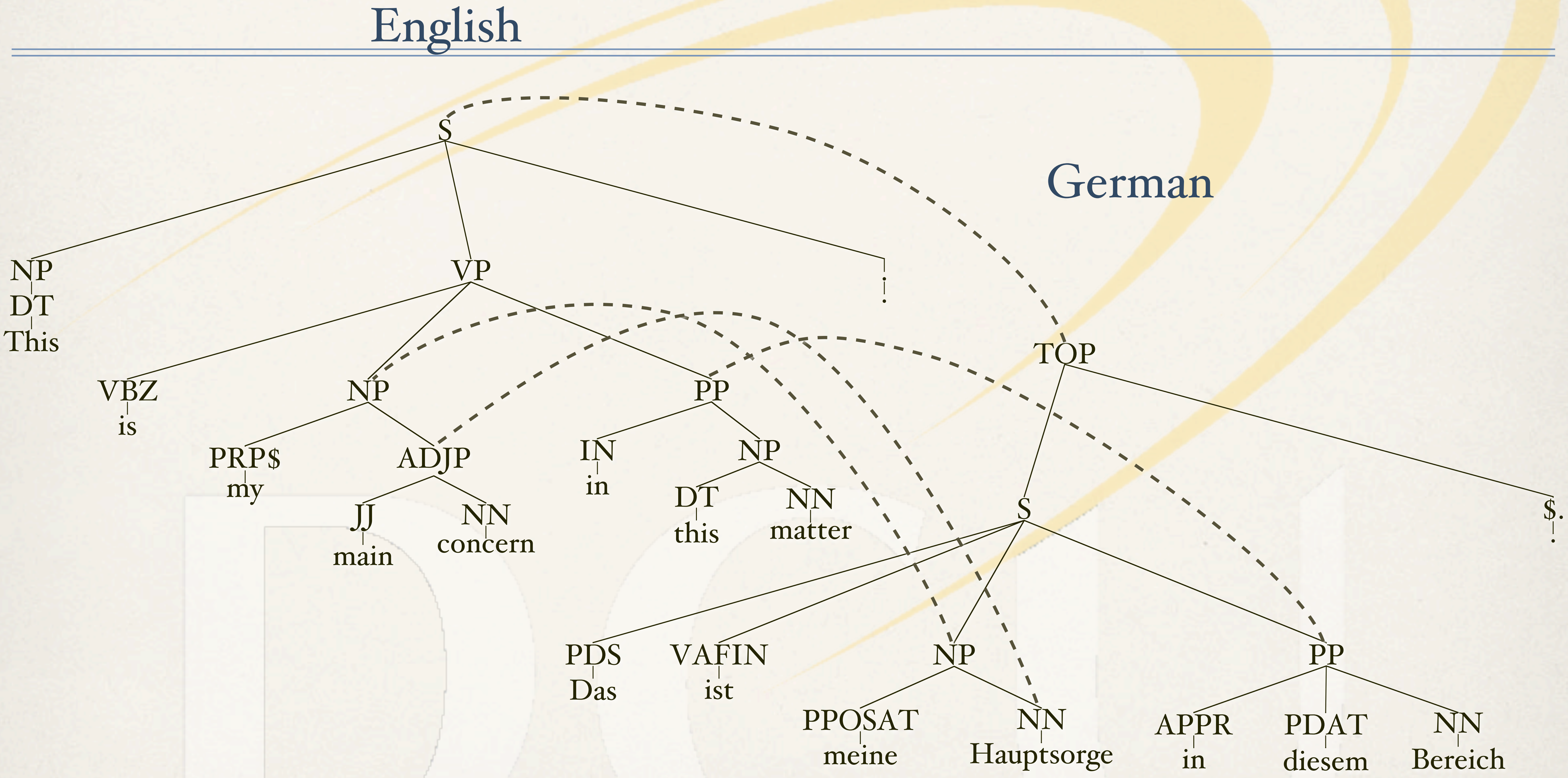
# Parallel Treebanks

| English | German |
|---|---|
| I do not think it is necessary for classic cars to be part of the directive . | Ich halte es nicht für notwendig , daß Oldtimer Bestandteil dieser Richtlinie sind . |
| I am not looking for such rigidly high recycling quotas when it comes to special-purpose vehicles either . | Auch bei Sonderfahrzeugen strebe ich nicht so unbedingt hohe Recyclingquoten an . |
| I want special-purpose vehicles such as ambulances to have high recovery quotas . | Ich habe den Wunsch , daß Sonderfahrzeuge wie Krankenwagen hohe Rettungsquoten haben . |
| This is my main concern in this matter . | Das ist meine Hauptsorge in diesem Bereich . |

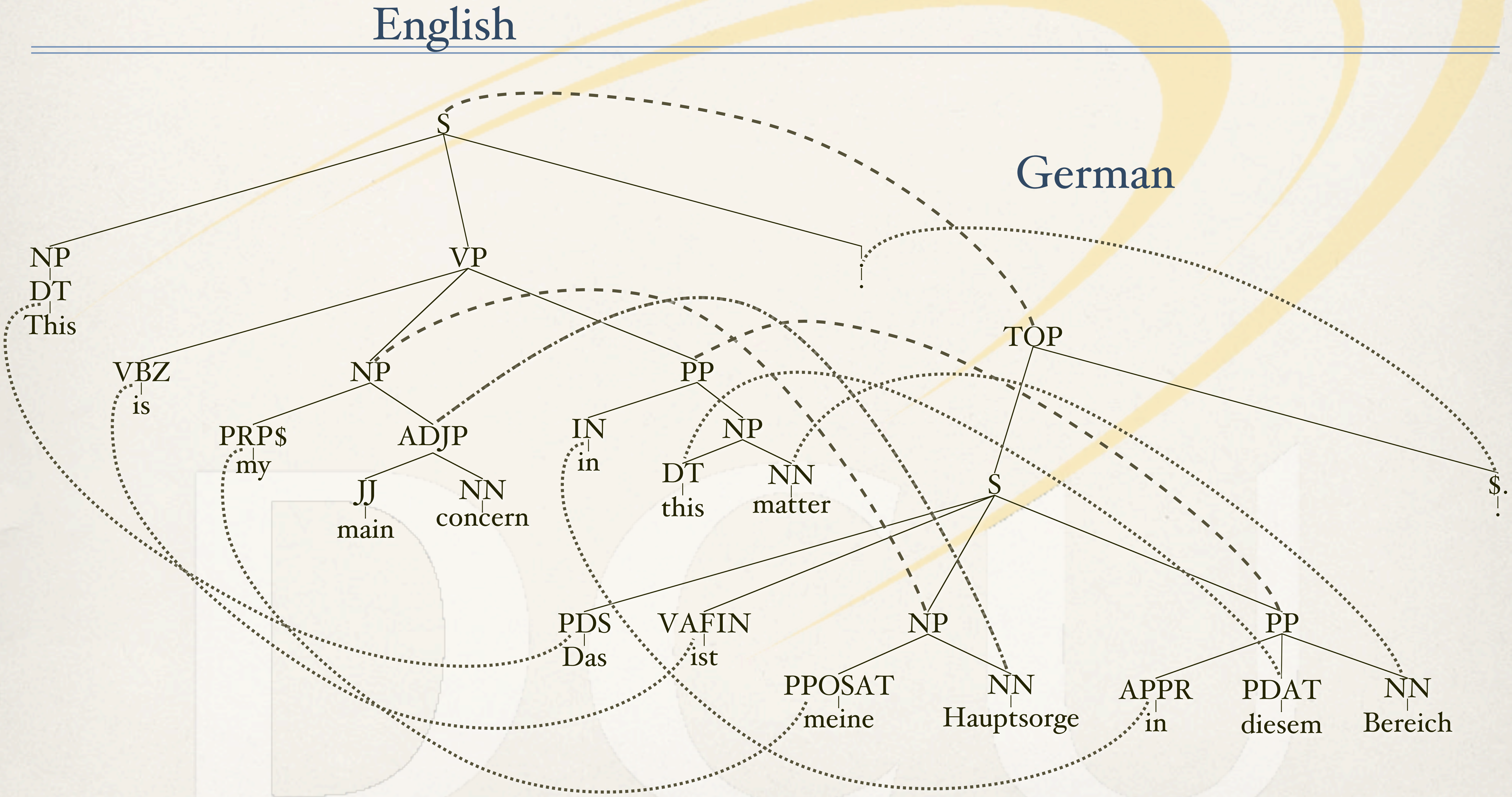
# Parallel Treebanks



# Parallel Treebanks



# Parallel Treebanks



# String-to-String Alignment

- The sub-tree aligner operates on parsed data
- For many languages no parsers are available
    - Retraining existing parsers for new languages may require significant resources
- The string-to-string aligner operates on plain sentences

# Alignment Algorithm

## *Bilingual Alignment*

- Align the SL fuzzy match to its TL translation from the TM
- The sub-tree aligner operates on plain unparsed data
- The probabilistic bilingual dictionary it uses may be generated using an off-the-shelf word-alignment tool (eg. GIZA++)

# Alignment Algorithm

## *Bilingual Alignment*

M

T

| X1 | X2 | X3 | X4 | X5 |
|---|---|---|---|---|
| sender | ’s | email | address | . |

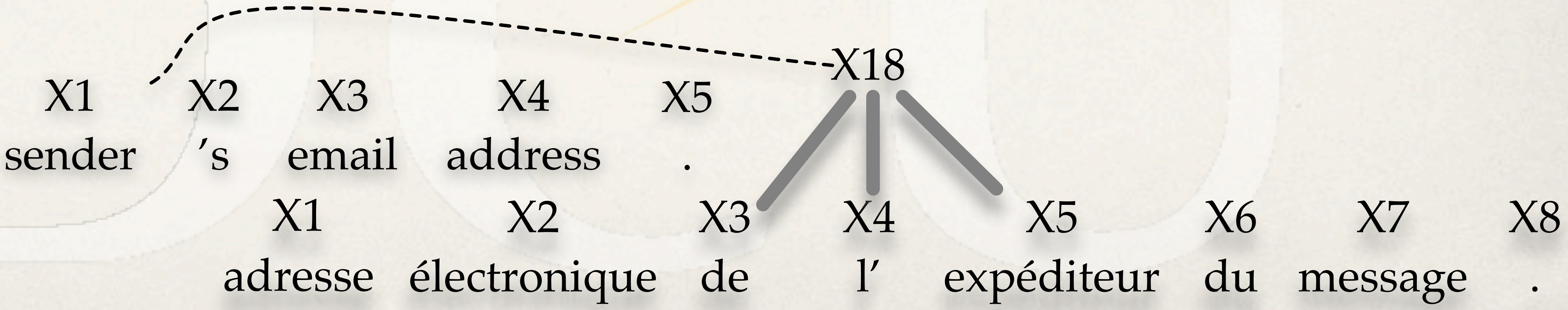
| X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 |
|---|---|---|---|---|---|---|---|
| adresse | électronique | de | l’ | expéditeur | du | message | . |

# Alignment Algorithm

## *Bilingual Alignment*

M

T



X18

X1 X2 X3 X4 X5

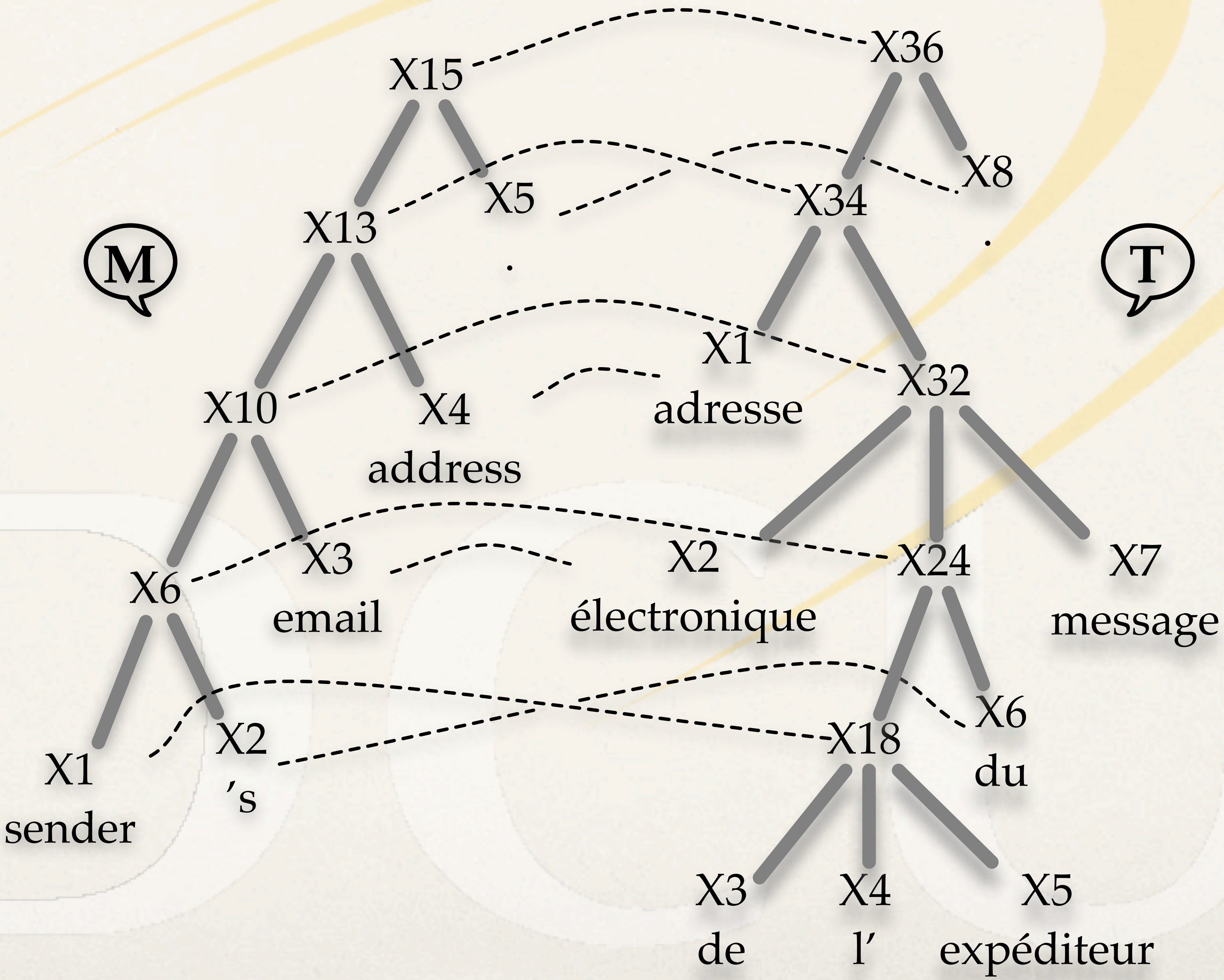
sender 's email address .

X1 X2 X3 X4 X5 X6 X7 X8

adresse électronique de l' expéditeur du message .

# Alignment Algorithm

## *Bilingual Alignment*



# Alignment Algorithm

## *Monolingual Alignment*

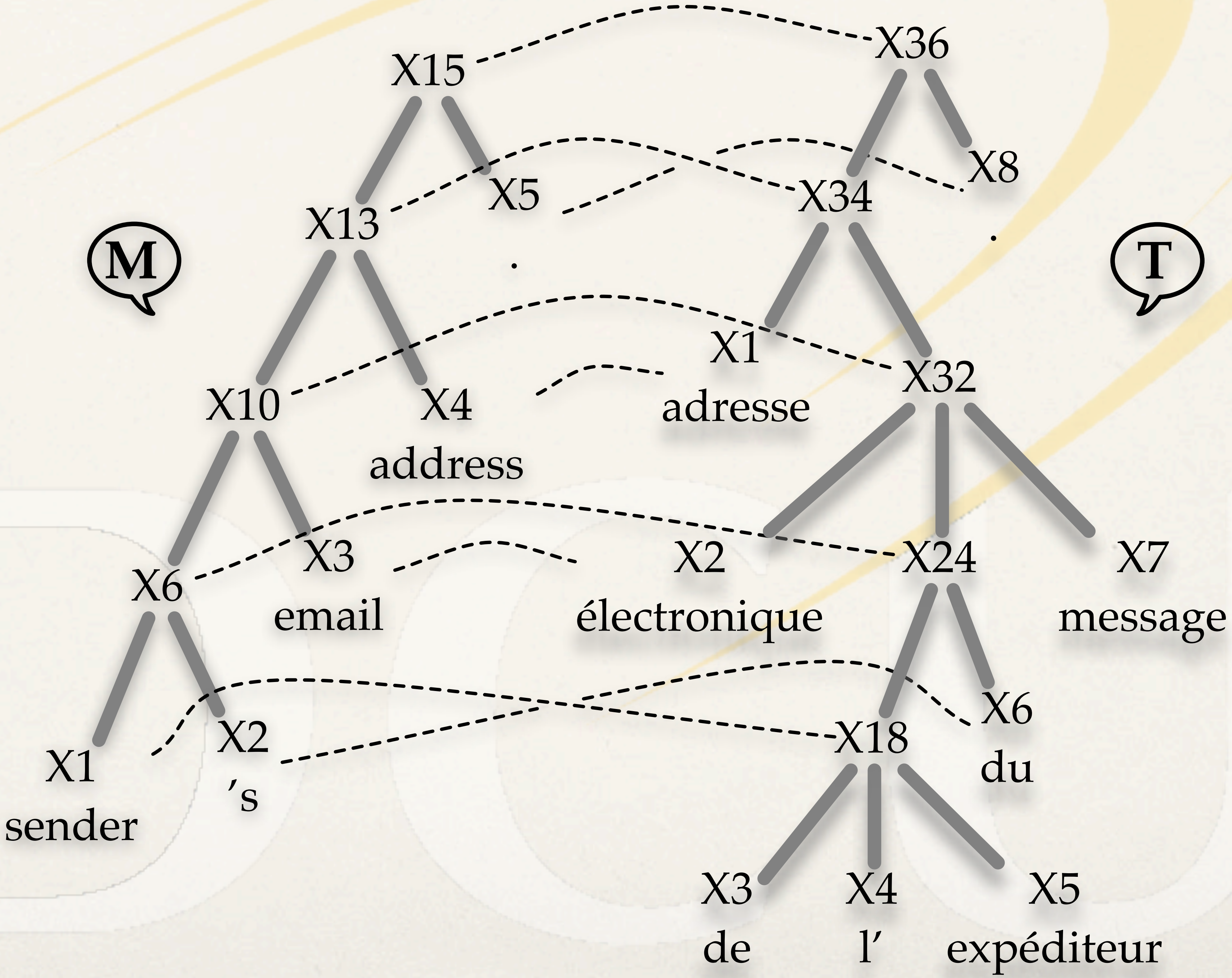
- Align the SL TM fuzzy match to the input sentence

- Namely, the plain input sentence to the structure derived for the SL TM fuzzy match during the *bilingual alignment*

- Use a dummy probabilistic dictionary, where each SL word available in the TM is aligned to itself with probability 1.

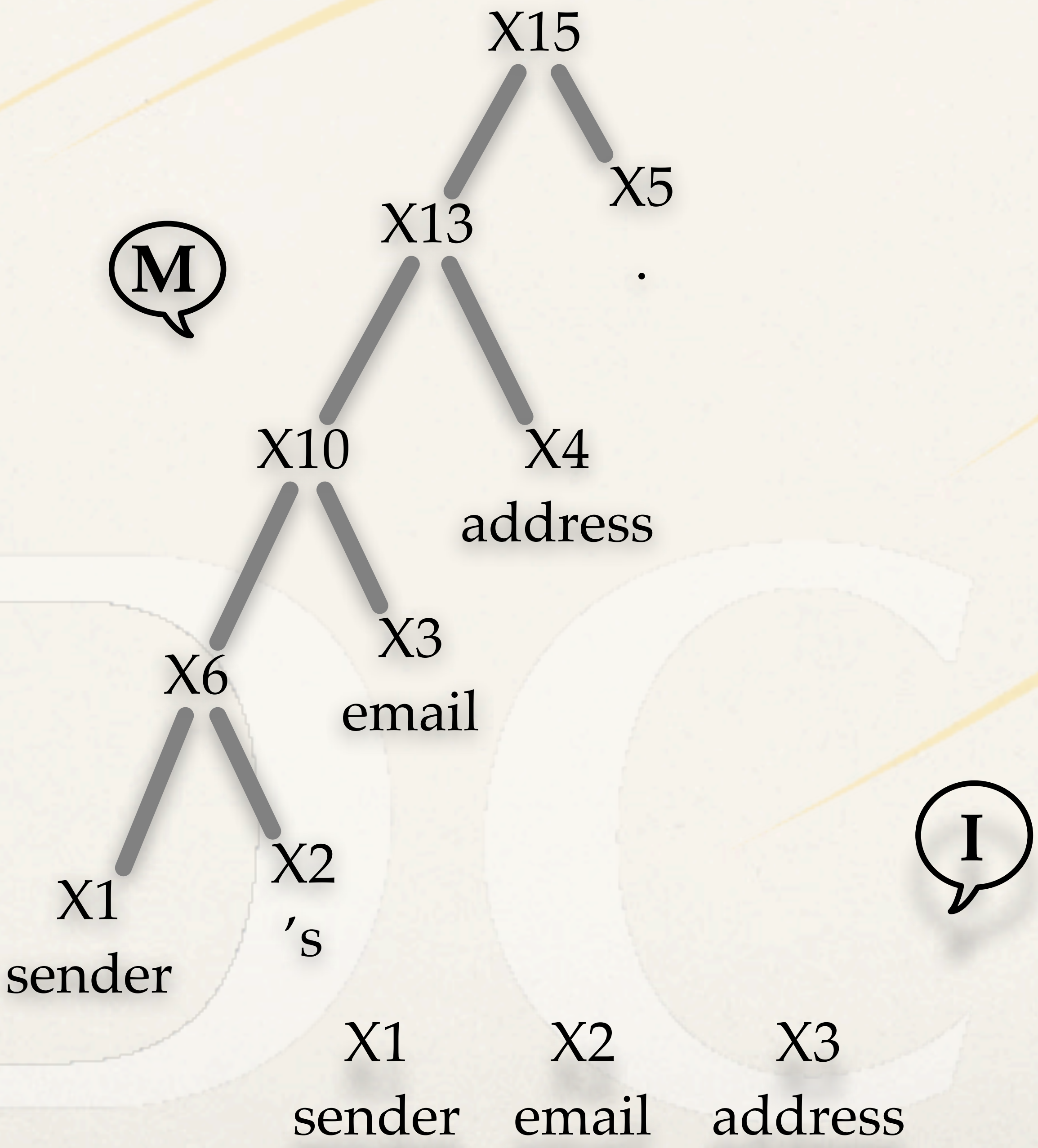
# Alignment Algorithm

*Monolingual Alignment*



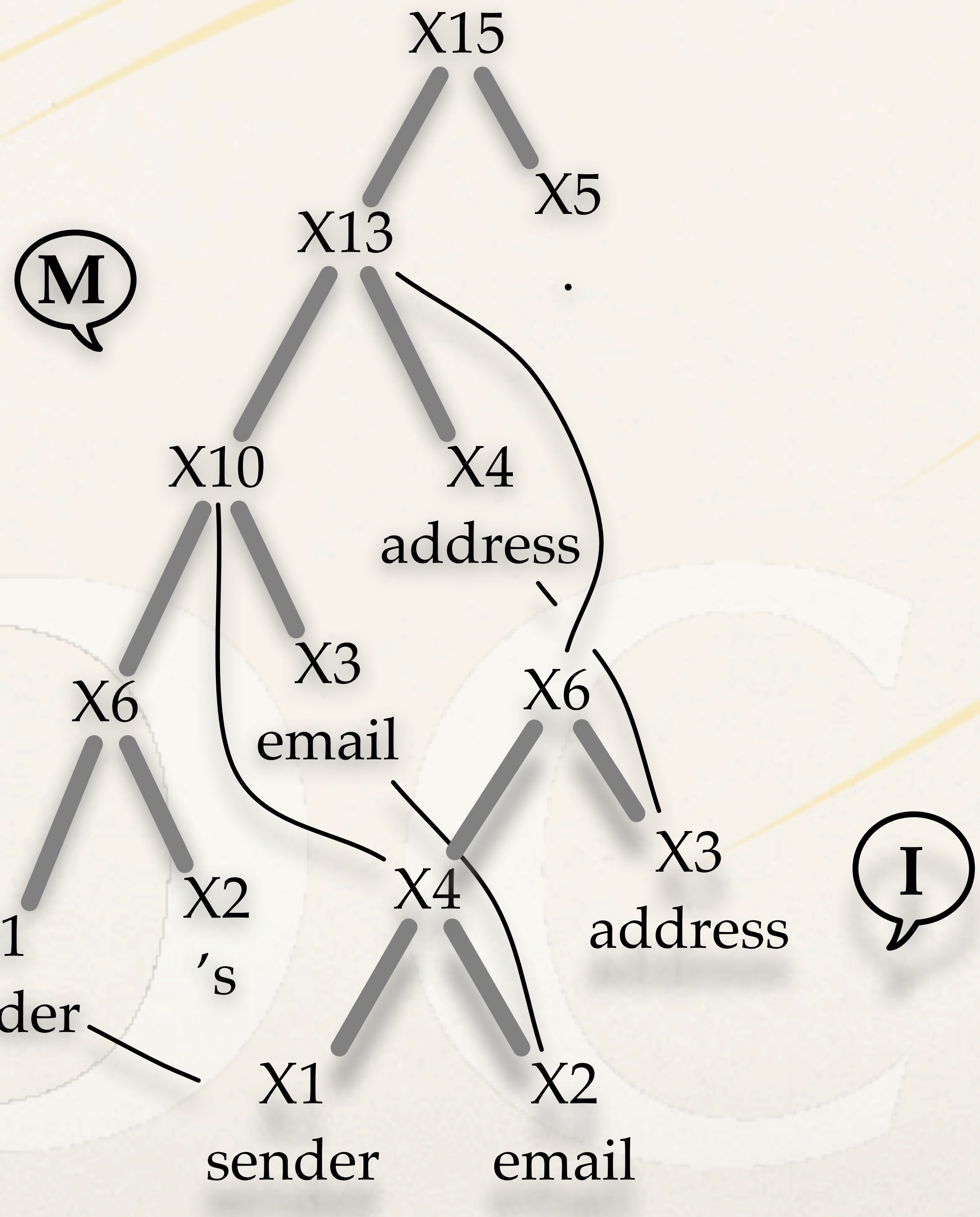
# Alignment Algorithm

*Monolingual Alignment*



# Alignment Algorithm

## *Monolingual Alignment*



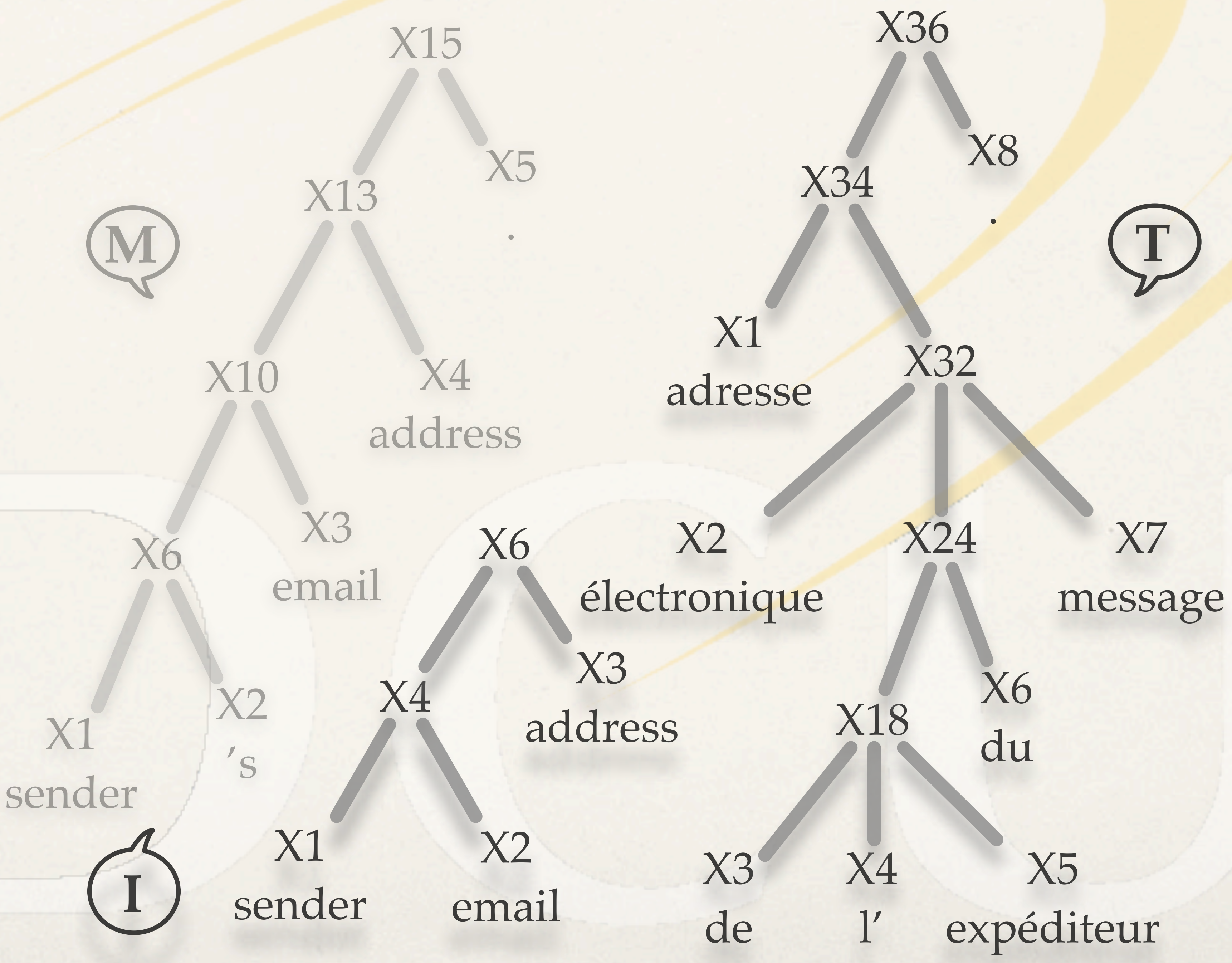
# Alignment Algorithm

## *Matching*

- The structure of the SL TM sentence is used as a pivot to align the structures of the input sentence and the TL TM sentence.

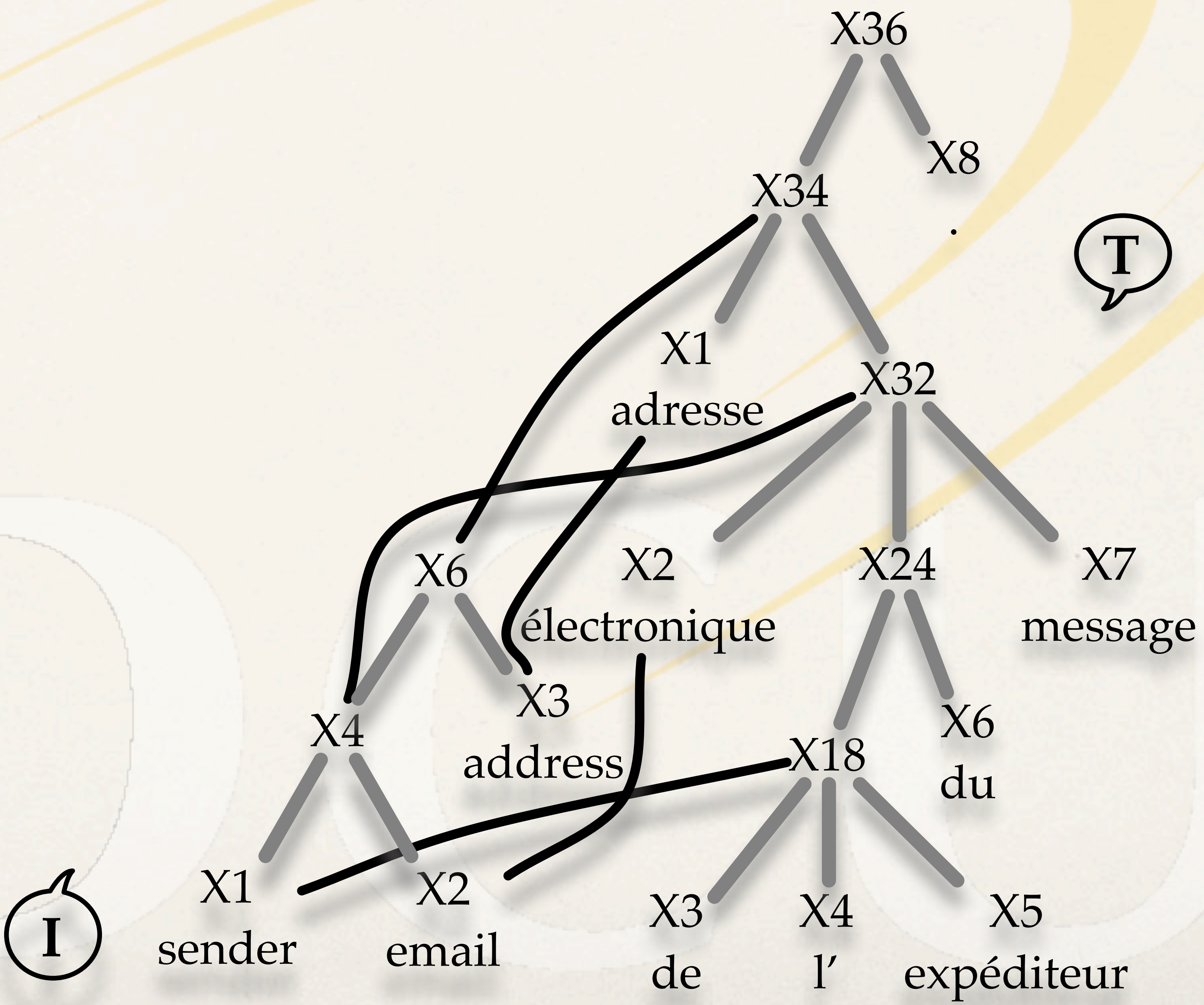
# Alignment Algorithm

*Matching*



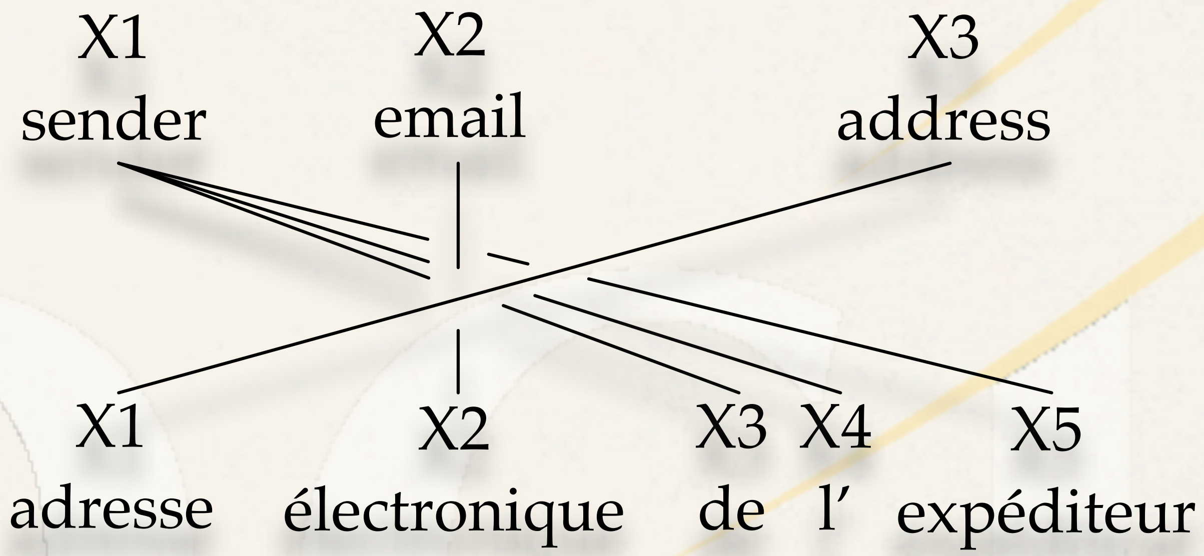
# Alignment Algorithm

*Matching*



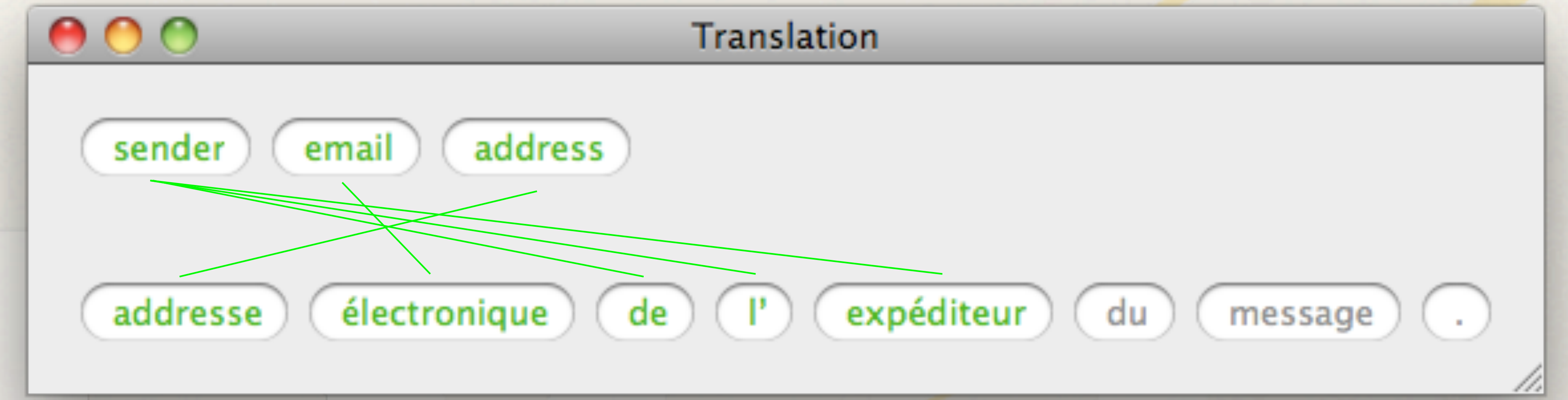
# Alignment Algorithm

## *Matching*



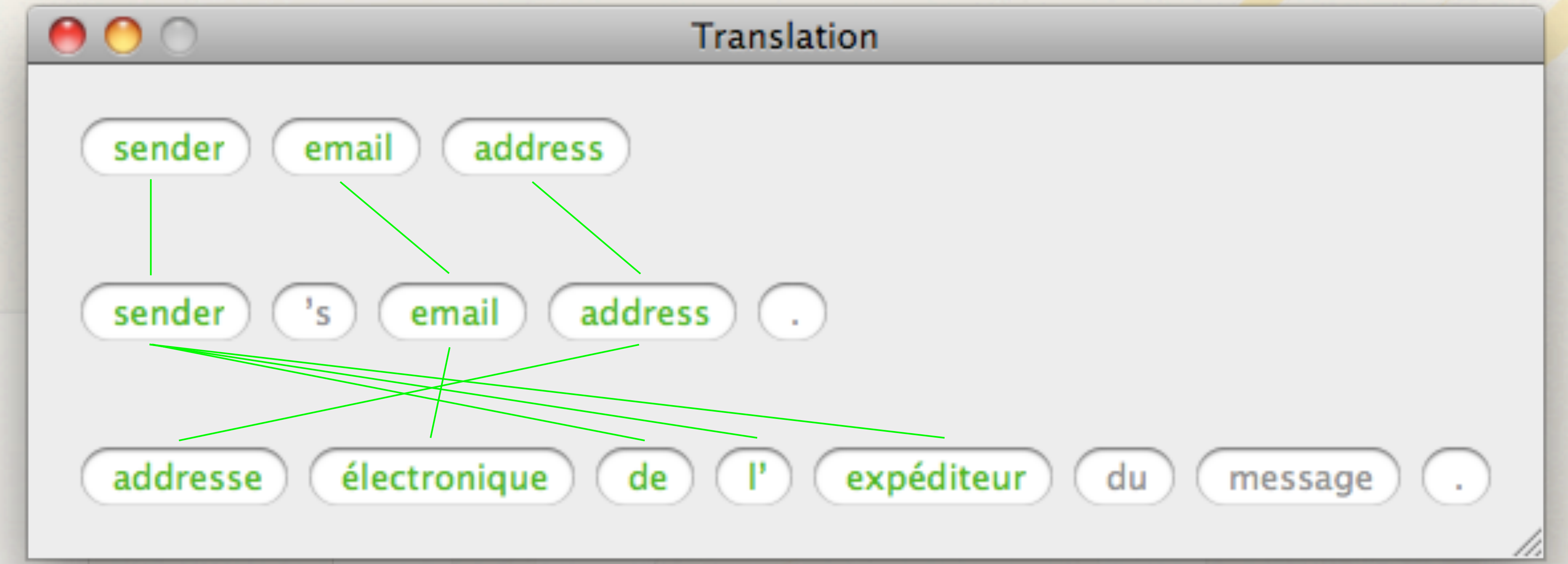
# Alignment Algorithm

*Matching*



# Alignment Algorithm

## *Matching*



# SMT Backend

- Use standard Moses for phrase-based SMT
- Two modes of operation:
  - ***comb*** translate the mismatched parts of the input individually using the SMT backend
  - ***xml*** mark-up the matched parts of the input with their translations and translate the marked-up input as a whole

# Reordering

- Use the parallel treebank to reorder the SL side of the TM to conform to the TL word order
- The SMT backend is then retrained to generate a ‘reordered’ model
- Both the regular and ‘reordered’ models are used during translation

# Evaluation Data

- Symantec EN–FR training data
  - 108 953 segment pairs
  - 13.2 EN average length
    15.0 FR average length
  - 41 379 EN unique tokens
    49 971 FR unique tokens

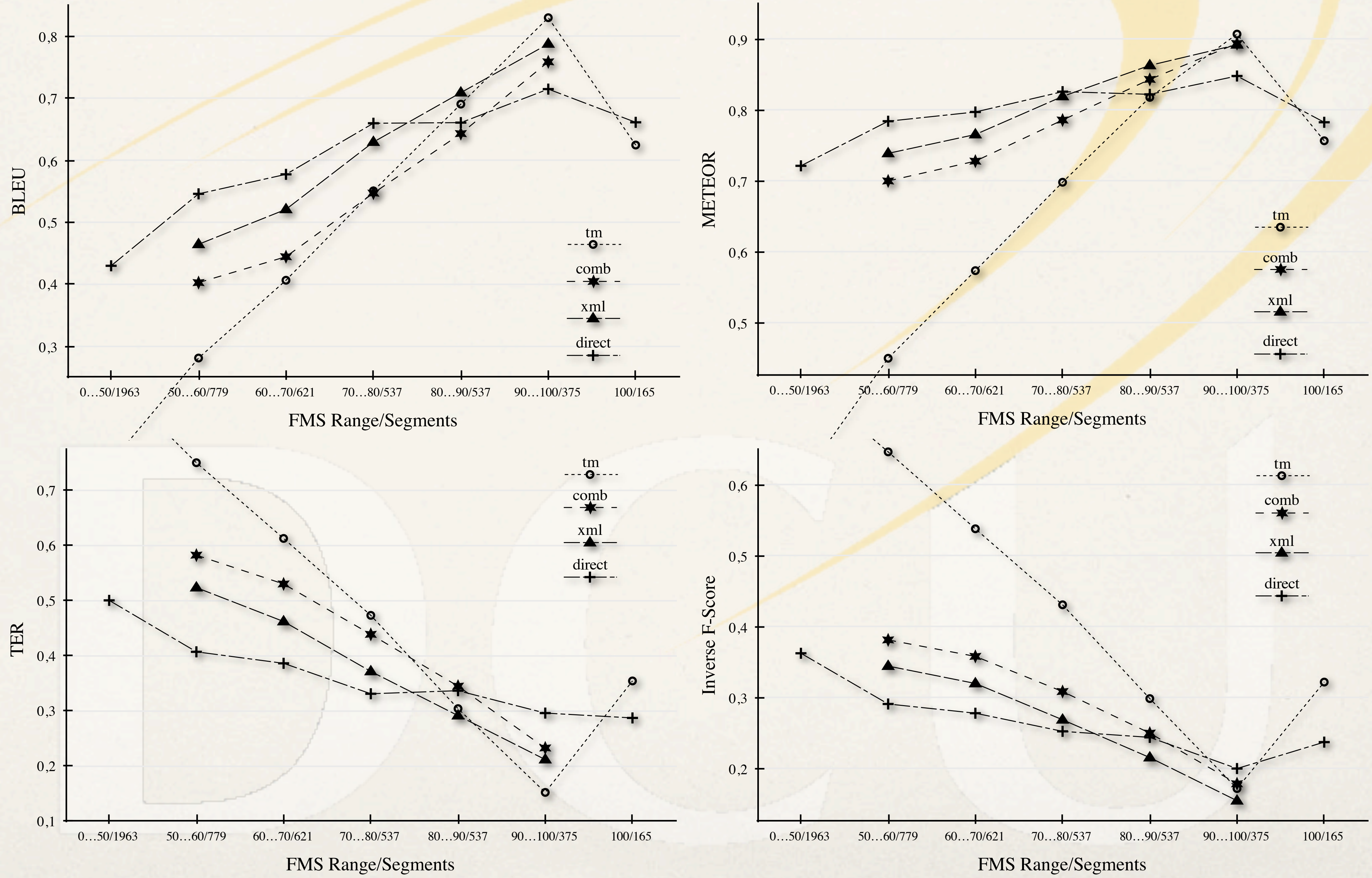
- Symantec EN–FR test data
  - 4 977 segment pairs
  - 9.2 EN average length
    10.9 FR average length

# Evaluation Data

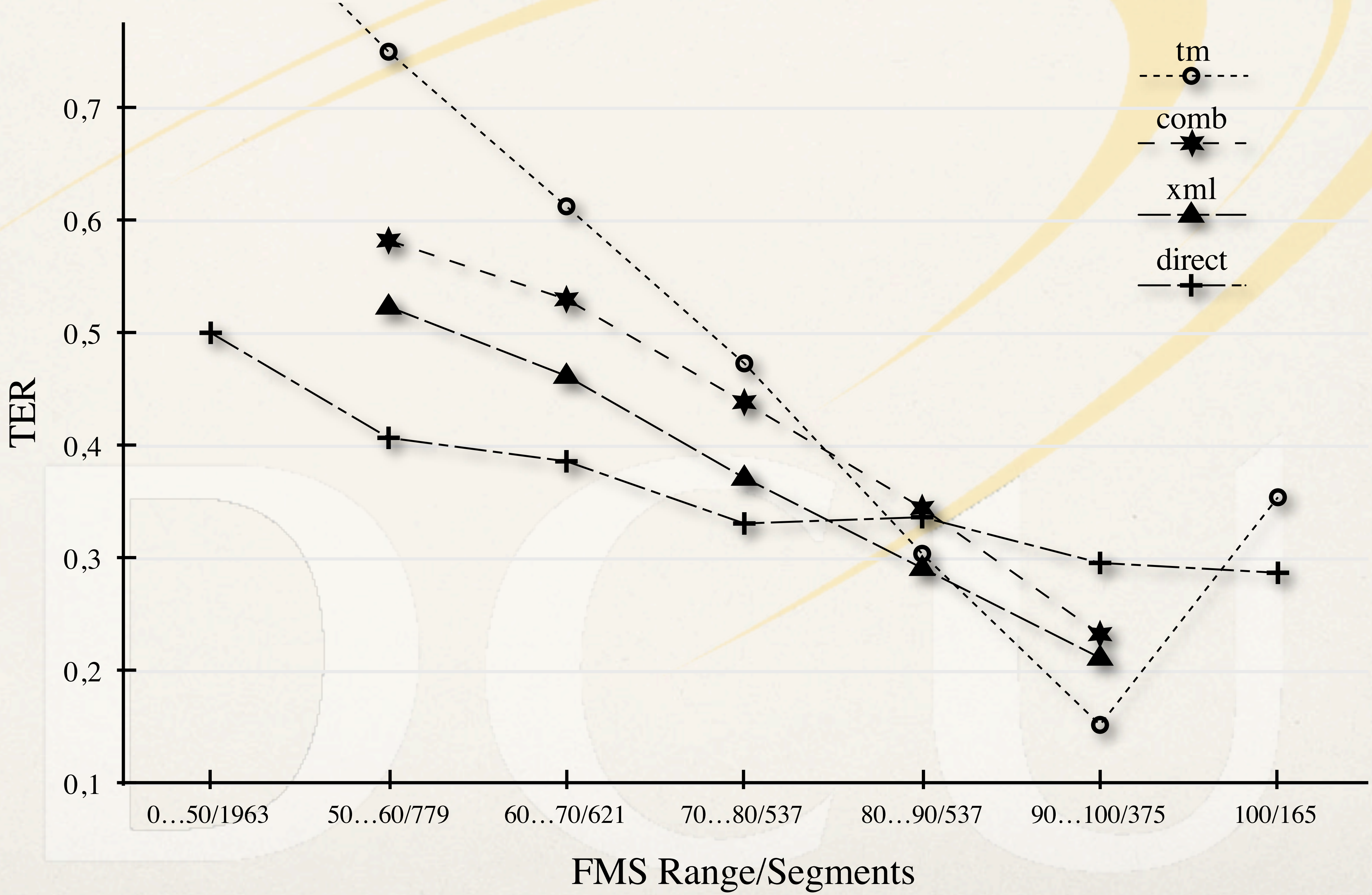
- Large number of XML tags
  - 2 049 EN unique tags
    2 653 FR unique tags
- Many 'special' strings
  - File paths
  - URLs
  - e-mail addresses
  - RTF formatting
  - XML tags with translatable parameters

- Meta-tag handling tool
- Specialised tokenizer

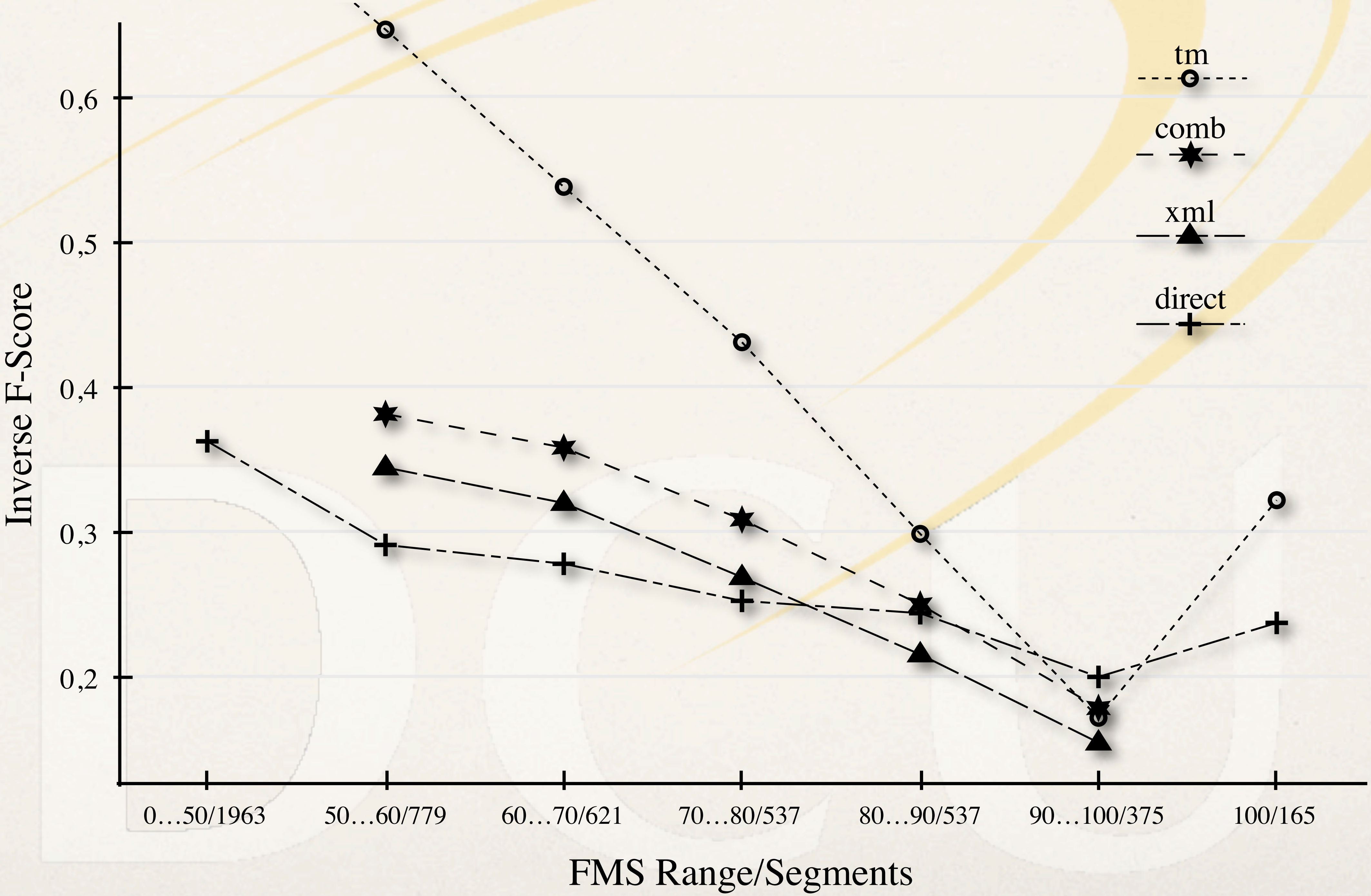
# Evaluation Results



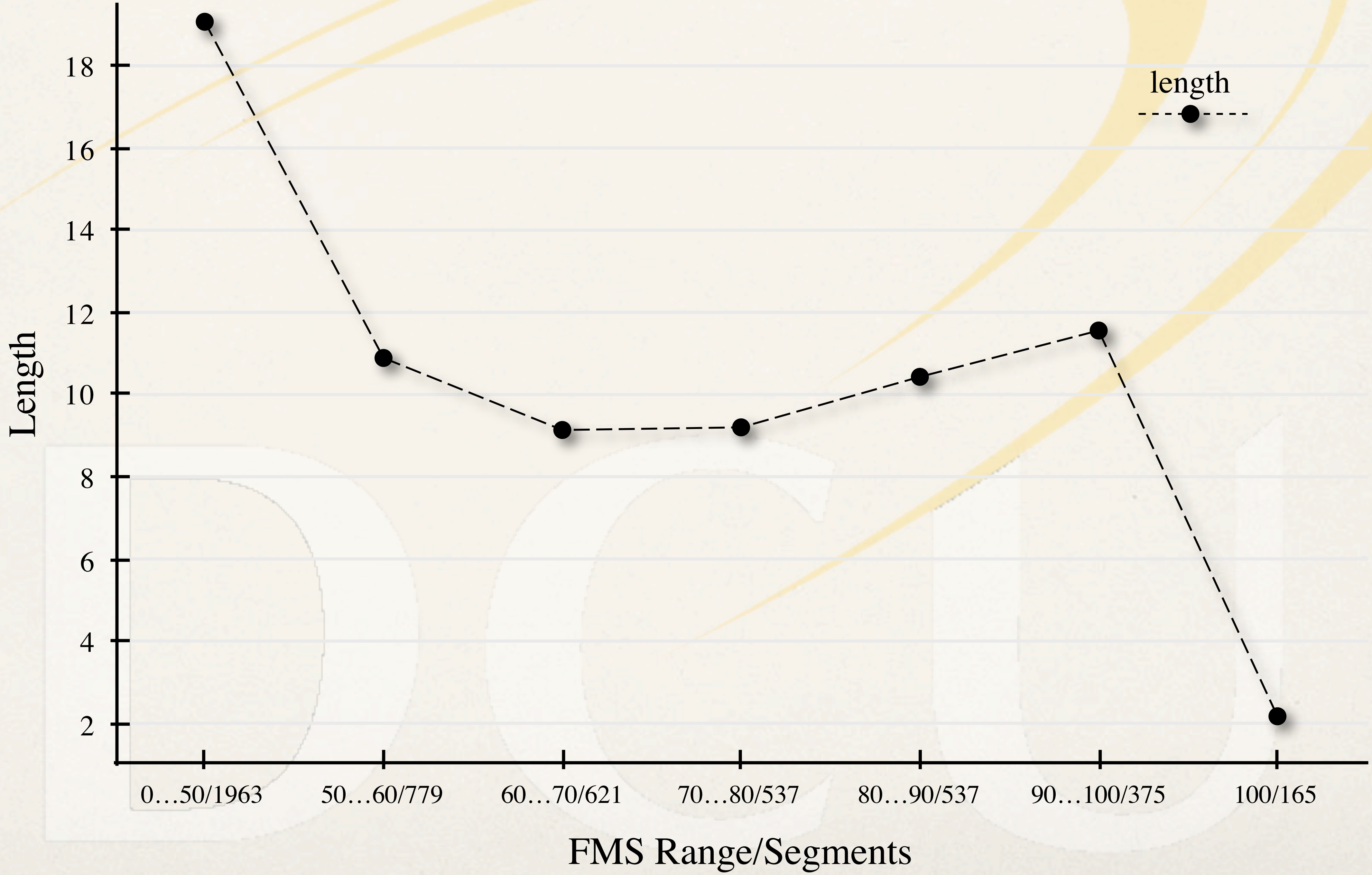
# Evaluation Results



# Evaluation Results



# Evaluation Results



# Future Work

- Develop a prototype implementation of the presented work
- Integrate this framework with a proper TM system
- Perform a user study to evaluate the effect of this framework on post-editing speed
- Further develop the meta-tag handling tool
  - possibly integrating it with the alignment and SMT backends
- Improve the reordering accuracy
- Run experiments where the SMT backend has been trained on additional data, besides the TM

# Thank you!

*http://VentsislavZhechev.eu*

*contact@VentsislavZhechev.eu*