Morphological Processing and Word Reordering for Statistical MT of Highly Inflected Languages

Marcello Federico  Arianna Bisazza  Christian Hardmeier
Human Language Technologies Research Unit
FBK-irst, Trento - Italy

Haifa, 24 January 2011
Overview

• Statistical MT in a nutshell
• When it works and when it does not
• Case study 1: Turkish to English
• Case study 2: Arabic to English
• Case study 3: German to English
• Conclusions

To take home:
embedding morpho-syntactic information into SMT is possible and it works!

This work was supported by the EuroMatrixPlus project (IST-231720), which is funded by the European Commission under the 7th Framework Programme for Research and Technological Development.
Freedom of movement must be encouraged, while ensuring that career paths are safeguarded.

E' necessario incoraggiare tale mobilità pur garantendo la sicurezza dei percorsi professionali.

**How SMT works** (in a nutshell)

- **operations**: segment, translate, and place
- **scores**: linear combination of feature functions
- **features**: phrase pairs, target n-grams, relative phrase movement, ...
- **search**: efficient algorithm to compute (sub-)optimal solutions
- **features and combination weights** are **machine learnable** from parallel data
Freedom of movement must be encouraged, while ensuring that career paths are safeguarded.

E’ necessario incoraggiare tale mobilità pur garantendo la sicurezza dei percorsi professionali.

When SMT works (when "more data" is not enough)

- simple morphology of source/target
  - better n-gram models, better alignments, less OOV words, ...

- similar morphology between source and target
  - better alignments, richer phrase tables, ...

- similar syntax between source and target
  - better alignments, phrase-tables, word re-ordering,...
For many language pairs we are far from the ideal condition.

What can we do? what has been done?

- **Enhance SMT features** to capture more information
  - factored models, shallow/deep syntax models, hierarchical re-ordering model
- **Integrate language knowledge** within the existing models
  - morphology pre-preprocessing, word-order pre-processing

We report recent work on the second approach for three translation directions:

- **Turkish** to English, IWSLT BTEC task
- **Arabic** to English, NIST MT 2009 task
- **German** to English, WMT 2010 task

All case studies are carried out with the Moses and IRSTLM toolkits.
Morphological Pre-processing for Turkish SMT


IWSLT BTEC Turkish-English task

Tourist expressions: simple task but limited training data
Rich morphology of Turkish has negative impact on SMT

<table>
<thead>
<tr>
<th>Training</th>
<th>OOV on Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>W</td>
</tr>
<tr>
<td>TR</td>
<td>139.5K</td>
</tr>
<tr>
<td>EN</td>
<td>182.6K</td>
</tr>
</tbody>
</table>

Examples:

SRC: Belki bir doktora görünmelisin.
REF: Perhaps you should see a doctor.
OUT: Maybe $[görünmelisin]$ a doctor.

SRC: Bu film rulolarını banyo ettirip basabilir miydiniz?
REF: Could you develop and print these rolls of film?
OUT: Could you reissue $[ettirip]$ $[rulolarını]$ this film developed?
Several linguistic features can negatively affect an SMT system:

- **Agglutination**
  → vocabulary built by a wide range of suffix combinations

  \[
  \begin{align*}
  \text{oda} & \quad \text{‘room’} \\
  \text{odam} & \quad \text{‘my room’} \\
  \text{odamda} & \quad \text{‘in my room’} \\
  \text{odamdayım} & \quad \text{‘I am in my room’}
  \end{align*}
  \]

- **Vowel harmony** and other phoneme alternation phenomena
  → systematic stem and suffix allomorphy

  \[
  \begin{align*}
  \text{saç} + (I)m & \rightarrow \text{saçım} \quad \text{‘my hair’} \\
  \text{el} + (I)m & \rightarrow \text{elim} \quad \text{‘my hand’} \\
  \text{kol} + (I)m & \rightarrow \text{kolum} \quad \text{‘my arm’} \\
  \text{göz} + (I)m & \rightarrow \text{gözüm} \quad \text{‘my eye’} \\
  \text{kafa} + (I)m & \rightarrow \text{kafam} \quad \text{‘my head’}
  \end{align*}
  \]
Turkish - Morphological Decomposition

**Idea:** selectively isolating or removing suffixes from the words

**Workflow:**

1. Morphological analysis and suffix normalization [Oflazer, 94]: suffix boundaries are detected and surface forms are replaced by tags to address vowel harmony and allomorphy.

2. Morphological disambiguation in context [Sak and Saraclar, 2007]: only the most likely analysis is taken for each word

3. Rules for splitting/removal of suffix tags:
   15 segmentation schemes developed and tested. Best performing schemes:
   - *MS11*: handles **nominal** suffixes (case, possessive) and copula;
   - *MS13*: also isolates **verbal** negation suffix;
   - *MS15*: also isolates other **verbal** suffixes: subject person, ability & voice.
Examples: surface form vs normalized representation:

*I was in my room*

= odamdaydım → oda / m / da / ydi / m

[room-my-in-was-l] [room] [my] [in] [was] [l]

*I can not explain*

= anlatamıyorum → anla / t / a / mı / yor / um

[understand-make-can-not-l] [understand] [make] [can] [not] [l]
Turkish - Morphological Decomposition

Examples: surface form vs normalized representation:

\[
\begin{align*}
I \text{ was in my room} &\rightarrow oda / m / da / ydi / m \\
\text{[room-my-in-was-I]} &\rightarrow \text{[room] [my] [in] [was] [I]} \\
\text{oda+A3sg/+P1sg/+Loc/+Zero+Past/+A1sg} &\uparrow \uparrow \uparrow \uparrow \uparrow \\
\text{lemma poss. case copula person} &\uparrow \uparrow \uparrow \uparrow \uparrow \\
\end{align*}
\]

\[
\begin{align*}
I \text{ can not explain} &\rightarrow anla / t / a / m\text{i} / yor / um \\
\text{[understand-make-can-not-I]} &\rightarrow \text{[understand] [make] [can] [not] [I]} \\
anla+Prog1/+Caus/+Able/+Neg/+A1sg &\uparrow \uparrow \uparrow \uparrow \uparrow \\
\text{lemma+tense causative ability negation person} &\uparrow \uparrow \uparrow \uparrow \uparrow \\
\end{align*}
\]

⇒ The underlying representation is used to train the SMT system.
**Turkish - Morphological Decomposition**

**Results:**
- minimizes differences in word granularity between TR and EN,
- abstracts from allomorphy by using a tag-like notation,
- reduces data sparseness, training dictionary size, OOV rate of test:

<table>
<thead>
<tr>
<th>Preprocessing</th>
<th>Training</th>
<th>OOV on Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
<td>W</td>
</tr>
<tr>
<td><strong>TR</strong> basic tokenization</td>
<td>139.5K 17.6K</td>
<td>6.7%</td>
</tr>
<tr>
<td>MS11</td>
<td>168.1K 10.4K</td>
<td>2.6%</td>
</tr>
<tr>
<td>MS15</td>
<td>174.5K 9.5K</td>
<td>2.0%</td>
</tr>
<tr>
<td><strong>EN</strong> basic tokenization</td>
<td>182.6K 8.3K</td>
<td>–</td>
</tr>
</tbody>
</table>
Turkish - Morphological Decomposition

• yields more refined alignments:

I'm with my girlfriend

I'm with my girlfriend

How do I get to this place?

How do I get to this place?
**Idea:** replace OOVs in the test by morphologically similar words seen in training:

- possible replacers: all words sharing the same lemma
- heuristic: choose candidates with tag sequence most similar to the OOV word
- OOVs replaced by $n$-best candidates in a confusion network input

<table>
<thead>
<tr>
<th>Word</th>
<th>Gloss</th>
<th>Preprocessed (MS11)</th>
<th>Score $^1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>çıkışlar</td>
<td>exits</td>
<td>çık+Verb+Pos^DB+Noun+Inf3+A3pl</td>
<td></td>
</tr>
<tr>
<td>çıkış</td>
<td>exit</td>
<td>çık+Verb+Pos^DB+Noun+Inf3+A3sg</td>
<td>93</td>
</tr>
<tr>
<td>çıkma</td>
<td>going out</td>
<td>çık+Verb+Pos^DB+Noun+Inf2+A3sg</td>
<td>66</td>
</tr>
<tr>
<td>çılacak</td>
<td>will go out</td>
<td>çık+Verb+Pos^DB+Noun+FutPart+A3sg</td>
<td>66</td>
</tr>
<tr>
<td>çıkan</td>
<td>who goes out</td>
<td>çık+Verb+Pos^DB+Adj+PresPart</td>
<td>44</td>
</tr>
<tr>
<td>çıkıyor</td>
<td>is going out</td>
<td>çık+Verb+Pos+Prog1</td>
<td>27</td>
</tr>
<tr>
<td>çıkmıyor</td>
<td>isn't going out</td>
<td>çık+Verb+Neg+Prog1</td>
<td>0</td>
</tr>
<tr>
<td>çıkarır</td>
<td>takes out</td>
<td>çık+Verb^DB+Verb+Caus+Pos+Aor</td>
<td>-15</td>
</tr>
</tbody>
</table>

$^1$Score $= 20C - 2D_1 - 5D_2$, where $C$: # of shared contiguous tags, $D_1$: # of different tags in the OOV, $D_2$: # of different tags in the candidate.
Segmentation lattice

- Choice of optimal decomposition ruleset depends on task & target language
- Possible approach: combine various degrees of decomposition in input
  \[\Rightarrow\] decoder can choose word-level-optimal segmentation path
- Training set = differently segmented versions of train, concatenated
- Example lattice combining MS11 + MS13 + MS15:

```
öksürük +P1sg
dur+Caus+Able+Neg+Prog1
+Neg
+Prog1
+Caus
+Able
+A1sg

TR: öksürüğümü durduramıyorum
(EN: I cannot make my cough stop)
```
## Experimental Results

<table>
<thead>
<tr>
<th>System</th>
<th>iwslt04</th>
<th>iwslt09</th>
<th>iwslt10</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>54.80</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>segm. ruleset MS11</td>
<td>60.30</td>
<td>57.21</td>
<td>52.14</td>
</tr>
<tr>
<td>segm. ruleset MS15</td>
<td>60.32</td>
<td><strong>58.28</strong></td>
<td>52.46</td>
</tr>
<tr>
<td>MS11 + lexical approx. (3-best)</td>
<td>59.68</td>
<td>57.11</td>
<td>51.76</td>
</tr>
<tr>
<td>segm. lattice MS11+13+15</td>
<td><strong>60.41</strong></td>
<td>57.70</td>
<td><strong>53.29</strong></td>
</tr>
</tbody>
</table>

- Morphological decomposition yields substantial improvements on baseline
- Adding rules for verbal inflection (MS15) helps slightly but consistently
- Lexical approximation unfortunately doesn’t help
- Decomposition lattice works best for two of the three test sets

**Conclusions**: choice of pre-processing technique depends on task and dataset.
Morphological Pre-processing for Arabic SMT

Rich morphology, but also orthographic variations and different vowelization styles. → specific preprocessing reduces data sparseness and improves alignments.

**Arabic tokenization:** Unicode characters and digits normalization, removal of diacritics and *tatweel* (justification character).

**Morphological decomposition:** isolates clitics from words.

Two state-of-the-art linguistic tools compared:

- **MADA**
  - heavy-weight: based on linguistic features produced by Buckwalter analyzer,
  - optimised use of the tool to run on large corpora

- **AMIRA**
  - light-weight: SVM classifier based on a $-5/+5$ character context.
Two different segmentation schemes:

- MADA (scheme D2) splits prefixes: conjunctions (w+ ‘and’, f+ ‘then’), prepositions (b+ ‘by’, k+ ‘as’, l+ ‘to’), future tense mark (s+). Also normalizes orthography (beginning alef, tah marbutah, alef maksura. . . )
- AMIRA doesn’t split future mark, but splits suffixes: object and poss. pronouns.

<table>
<thead>
<tr>
<th>Baseline</th>
<th>MADA</th>
<th>AMIRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>wstqw lh</td>
<td>w+ s+ twq lh</td>
<td>w+ stqwl +h l+ zmyl +hA</td>
</tr>
<tr>
<td>lzmylh A</td>
<td>l+ zmy lh A</td>
<td>[and-she-will-say-it] [to-her-colleague]</td>
</tr>
<tr>
<td>[and-she-will-say-it] [to-her-colleague]</td>
<td>[and] [will] [she-say-it] [to] [her-colleague]</td>
<td></td>
</tr>
<tr>
<td>‘and she will say it to her colleague’:</td>
<td>[and] [she-will-say] [it] [to] [colleague] [her]</td>
<td></td>
</tr>
</tbody>
</table>

On the NIST task MADA slightly outperforms AMIRA, but AMIRA is faster and includes shallow chunking.
Verb Reordering for Arabic SMT


The Problem of Arabic VSO Sentences

**Problem:**
Word reordering is a challenge for phrase-based SMT between distant languages
English: mainly Subject-Verb-Object VS Arabic: both SVO and VSO

Typical errors in phrase-based SMT outputs:

*The Moroccan monarch King Mohamed VI renewed his support to the French President*
First attempt: rule-based verb reordering

- Focus on verbs *say, declare, note*... in pre-subject position of news
- Apply simple surface pattern-matching reordering rules, without syntax
- Rule: move *verb* before *trigger element* (*‘that’, colon, quotation mark, etc.*)

**Example 1**

**original**

src: \(q\text{Alt}\text{ hh AlwkAlp : nZrA l+ AlwDE AlHAly fy AlErAq} \ldots\)

mt: *She said* the agency: *In view of the current situation in Iraq* \ldots

**reordered**

src: \(h*h\text{ AlwkAlp qAlt : nZrA l+ AlwDE AlHAly fy AlErAq} \ldots\)

mt: *The agency said* due to the current situation in Iraq \ldots
First attempt: rule-based verb reordering

- Focus on verbs *say, declare, note...* in pre-subject position of news
- Apply simple surface pattern-matching reordering rules, without syntax
- Rule: move verb before *trigger element* (‘that’, colon, quotation mark, etc.)

**Example 2**

**original**

src:  tAbE byAn SAdr En mktb hnyp >n A1>xyr ...
mt:  He went on to say, a statement issued by the office of Hania that the latter

**reordered**

src:  byAn SAdr En mktb hnyp tAbE >n A1>xyr ...
mt:  A statement issued by the office of Hania continued that the latter ...
First attempt: rule-based verb reordering

- Focus on verbs *say, declare, note*... in pre-subject position of news
- Apply simple surface pattern-matching reordering rules, without syntax
- Rule: move verb before trigger element (‘that’, colon, quotation mark, etc.)

**Example 2**

**original**

src:  tAbE byAn SAdr En mktb hnyp >n Al>xyr ...
mt:  He went on to say, a statement issued by the office of Hania that the latter

**reordered**

src:  byAn SAdr En mktb hnyp tAbE >n Al>xyr ...
mt:  A statement issued by the office of Hania continued that the latter ...

Unfortunately, no significant BLEU improvement on the NIST task. We introduce more linguistic knowledge and extend to all verbs!
Assumptions:
1) verb reordering only between shallow syntax chunks
2) no overlap between consecutive verb movements

Define a class of possible movements:
i) move verb chunk...

ii) ... or verb chunk + next chunk (e.g. adverbials) by up to N chunks to the right

Best movement in parallel corpus: minimizes global distortion wrt to English translation
The reordered parallel corpus is used to train the SMT system. As for the test, we use word **reordering lattices**.

Given the initial assumptions, we can build compact lattices and run non-monotonic decoding on them (Dyer & al. 2008).

Hybrid approach:
- for verb reordering: lattices
- for other reorderings: standard (phrase-internal and local distortion)

Lattice representation of the rule:
“move 1 or 2 chunks by up to 6 chunk positions right”
Evaluation

High-end baseline: Moses, 30M words newswire from NIST09 with lexicalized reordering models (Och & al. 2004; Koehn & al. 2007)

Different experimental conditions:

- whole system re-trained and tuned on verb-reordered data
- translation of plain input (text)
- translation of reordering lattice

<table>
<thead>
<tr>
<th>System</th>
<th>DL</th>
<th>Eval08-NW</th>
<th>Eval09-NW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>bleu</td>
<td>krs²</td>
</tr>
<tr>
<td>baseline</td>
<td>6</td>
<td>43.10</td>
<td>80.57</td>
</tr>
<tr>
<td>reord. training +</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>plain input</td>
<td>6</td>
<td>43.67</td>
<td>80.62</td>
</tr>
<tr>
<td>lattice</td>
<td>4</td>
<td>44.04</td>
<td>80.93</td>
</tr>
<tr>
<td>oracle reordered</td>
<td>4</td>
<td>44.36</td>
<td>81.29</td>
</tr>
</tbody>
</table>

¹Kendall Reordering Score: similarity btw word order of outputs and of references (Birch & al. 2010)
We use **syntactic tree kernel** to represent verb chunk movements; Fig. shows forest corresponding to one specific movement. We train a SVM by optimizing global distortion in the training data.
### Conclusions

<table>
<thead>
<tr>
<th>System</th>
<th>DL</th>
<th>Eval08-NW bleu</th>
<th>Eval08-NW krs</th>
<th>Eval09-NW bleu</th>
<th>Eval09-NW krs</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>6</td>
<td>43.10</td>
<td>80.57</td>
<td>48.13</td>
<td>83.17</td>
</tr>
<tr>
<td>reord. training +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>full lattice</td>
<td>4</td>
<td>44.04</td>
<td>80.93</td>
<td>48.96</td>
<td>83.75</td>
</tr>
<tr>
<td>1-best-pruned</td>
<td>4</td>
<td>44.34</td>
<td>81.18</td>
<td>49.10</td>
<td><strong>84.15</strong></td>
</tr>
<tr>
<td>2-best-pruned</td>
<td>4</td>
<td>44.29</td>
<td><strong>81.30</strong></td>
<td>49.19</td>
<td>84.02</td>
</tr>
<tr>
<td>3-best-pruned</td>
<td>4</td>
<td>44.11</td>
<td>81.13</td>
<td>49.05</td>
<td>83.90</td>
</tr>
</tbody>
</table>

- Simply reordering of the training data is beneficial: more monotone alignments ⇒ better phrase extraction
- Providing likely reordering in the lattice yields further improvement
- Interesting: reordering-specific metric correlates well with BLEU
- Further improvement:
  - pruning the lattice with discriminative approach (SVM)
Morphological Reduction and Reordering for German

Morphology

- Inflectional morphology: much more prolific in German
  Nouns have case, verbs have many forms, etc.

- Derivational morphology:
  German has one-word-compounds that must be split

→ many vocabulary types, high OOV rate

Word order

- English: strict SVO word order
- German: SVO in main clauses, SOV in subordinate clauses

→ word order mismatch

Approach: morphological reduction and chunk-based reordering
We use Gertwol to split compounds and reduce words to their base form. Gertwol: commercial two-level finite-state morphology.

Gertwol analyses are disambiguated with POS tags and heuristic disambiguation rules (courtesy of the University of Zurich).

Decoding: supply reduced forms as alternative paths in a lattice:

Training: concatenate original and processed parallel texts.

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dev</td>
<td>Eval</td>
</tr>
<tr>
<td>Baseline</td>
<td>18.8</td>
</tr>
<tr>
<td>with morphological reduction</td>
<td>19.3</td>
</tr>
<tr>
<td></td>
<td>20.1</td>
</tr>
<tr>
<td></td>
<td>20.6</td>
</tr>
</tbody>
</table>
Chunk Reordering

- Same mechanism as for Arabic-English, but different rules.
- We concentrate on a few patterns involving verbs.
- Simplifying assumption:
  Verb reordering only occurs between shallow syntax chunks.
- Tagging and chunking done with the TreeTagger.
- Small number of hand-written reordering rules that can generate multiple reorderings for each matching verb chunk.

Example: Subordinate clause rule

Motivation: Move clause-final verbs in German SOV subordinates left to match English SVO word order.

Moving block: Verb chunk immediately followed by punctuation.

Movement: to the left
1 to 3 chunks after most recent subordinating conjunction

It is straightforward to merge a morphological reduction lattice with a chunk reordering lattice:

English-German: Results

<table>
<thead>
<tr>
<th>BLEU</th>
<th>DEV</th>
<th>EVAL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MR</td>
<td>MR</td>
</tr>
<tr>
<td>Baseline</td>
<td>18.8</td>
<td>19.3</td>
</tr>
<tr>
<td>with reordering</td>
<td>18.9</td>
<td>19.8</td>
</tr>
</tbody>
</table>

\[ MR = \text{morphological reduction} \]

- Chunk reordering on its own helps very little: around 0.2 BLEU points.
- In combination with morphological reduction, the gain is much greater: half a point for morphological reduction + half a point for reordering = one point total improvement
- Reordering with lattices strongly depends on the language model to choose the right path.
Conclusions

• We showed methods to exploit morpho-syntactic information for SMT – that also resulted in performance improvements on strong baselines

• Language expertise of the source/target languages definitely helps – to identify, analyze, and describe issues from a linguistic perspective

• Statistical modeling expertise is required
  – to conceive, implement, and integrate new features in the decoder
  – to exploit or extend existing features

• The phrase-based SMT framework is simple, flexible, and extensible
  – there are more and more things that can be explored, improved, integrated

• Current evolution of the presented approaches:
  – re-ordering models embedding language specific syntactic constraints/preferences
  – context models to enforce cohesive MT across different sentences
Conclusions

• We showed methods to exploit morpho-syntactic information for SMT – that also resulted in performance improvements on strong baselines.

• Language expertise of the source/target languages definitely helps – to identify, analyze, and describe issues from a linguistic perspective.

• Statistical modeling expertise is required – to conceive, implement, and integrate new features in the decoder – to exploit or extend existing features.

• The phrase-based SMT framework is simple, flexible, and extensible – there are more and more things that can be explored, improved, integrated.

• Current evolution of the presented approaches:
  – integrate language-specific word-order knowledge directly in the decoder
  – embed syntactic knowledge in re-ordering models and future cost estimation.

thank you