System Combination Using Joint, Binarised Feature Vectors

Christian Federmann, DFKI LT Lab

META RESEARCH

Overview

- Motivation
- Methodology
- Experiments
- Results
- Conclusion

translation 1
translation 2
translation 3
translation 4
translation 5

Motivation

Machine Translation

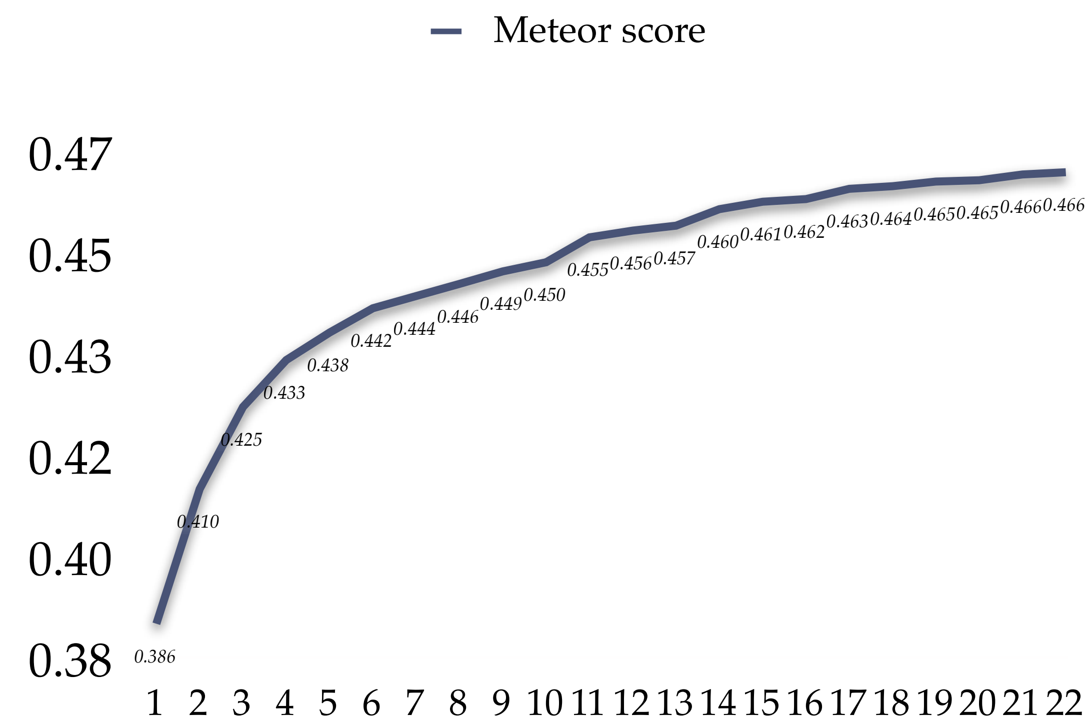
- Machine translation is a complex problem
- Several paradigms co-exist, each having individual strengths and weaknesses, e.g.:
  - Statistical Machine Translation (SMT)
  - Rule-based Machine Translation (RBMT)
- Possible solution: Hybrid Machine Translation

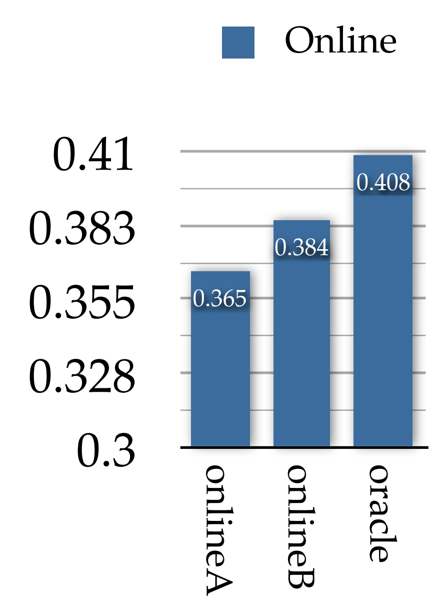
Hybrid MT

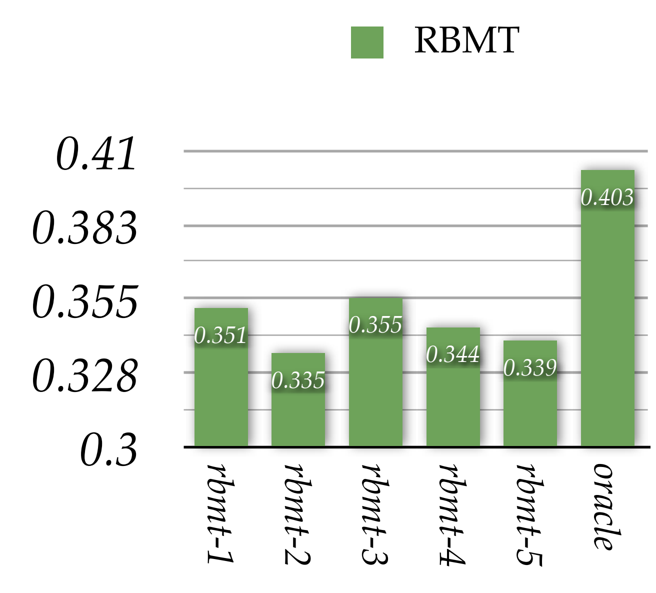
- Focuses on creation of combined translations
- Assumes that systems have individual, often complementary, strengths and weaknesses
- Clever combination of translations should result in an improved translation
- ML4HMT-11/-12 specifically investigate this :)

Oracle Scores

- Oracle experiments with WMT'11 translation data
- Good translations found for all translation systems
- Proposed approach better than combo systems
- Improvements regardless of specific language pair







MT + Machine Learning

- MT systems use a lot of heterogeneous features
- Simple scores, probabilities, or even parse trees
- Very difficult to *intuitively* understand systems
- Machine Learning techniques can help here

Methodology

Combination Approach

translation 1
translation 2
translation 3
translation 4
translation 5

translation 1
translation 2
translation 3
translation 4
translation 5

translation 1
translation 2
translation 3
translation 4
translation 5

↓

translation 1
translation 2
translation 3
translation 4
translation 5

Pick best translation #1

translation 1	translation 1	translation 1
translation 2	translation 2	translation 2
translation 3	translation 3	translation 3
translation 4	translation 4	translation 4
translation 5	translation 5	translation 5

↓

translation 1
translation 2
translation 3
translation 4
translation 5

Pick best translation #2

translation 1
translation 2
translation 3
translation 4
translation 5

translation 1
translation 2
translation 3
translation 4
translation 5

translation 1
<u>translation 2</u>
translation 3
translation 4
translation 5

↓

translation 1
translation 2
translation 3
translation 4
translation 5

Pick best translation #3

translation 1
translation 2
<u>translation 3</u>
translation 4
translation 5

translation 1
translation 2
translation 3
translation 4
translation 5

translation 1
translation 2
translation 3
translation 4
translation 5

translation 1
translation 2
translation 3
translation 4
translation 5

Pick best translation #4

translation 1
translation 2
translation 3
<u>translation 4</u>
translation 5

translation 1
translation 2
translation 3
translation 4
translation 5

translation 1
translation 2
translation 3
translation 4
translation 5

↓

translation 1
translation 2
translation 3
translation 4
translation 5

Pick best translation #5

translation 1 translation 2 translation 3 translation 4 translation 5	translation 1 translation 2 translation 3 translation 4 translation 5	translation 1 translation 2 translation 3 translation 4 translation 5

↓

translation 1
translation 2
translation 3
translation 4
translation 5

Requirements

- Mechanism to select locally best translation
  - Total order on translation output
  - Feature vectors modeling comparison
- Definition of a suitable set of features
- Training of a SVM-based classification model
- System combination with conflict resolution

Methodology

- n translations from several, black-box systems
- Training data includes source text and reference
- Decompose into pairwise A, B comparisons
- Round-robin tournament for sentence selection

Total Order

- Translation quality estimated using a multi-level, total order `ord(A, B)`
- Preference for sentence-based scores: Meteor
- Fallback to corpus-based metrics Meteor, NIST and BLEU, if necessary
- Extension with human judgment possible

"Classical" Features

- number of target tokens, parse tree nodes, and parse tree depth;
- ratio of target/source tokens, parse tree nodes, and parse tree depth;
- n-gram score for n-gram order $n \in \{1, ..., 5\}$;
- perplexity for n-gram order $n \in \{1, ..., 5\}$.

Individual Feature Vectors

$$vec_{single}(A) \stackrel{\text{def}}{=} \begin{pmatrix} f_1(A) \\ \vdots \\ f_n(A) \end{pmatrix} \in \mathbb{R}^n$$

- Quality estimation for MT usually based on feature vectors for single systems
- Classifier output is then combined in *some* way

Joint, Binarised Feature Vectors

$$vec_{binarised}(A, B) \stackrel{\text{def}}{=} \begin{pmatrix} f_1(A) > f_1(B) \\ \vdots \\ f_n(A) > f_n(B) \end{pmatrix} \in \mathbb{B}^n$$

- We use a different strategy, defining feature vectors which *explicitly* compare two systems
- Feature values are now compared as *"A>B?"*

Selection Mechanism

translation 1

translation 1

translation 1

↓

???

ord(X,Y) can only be approximated!

Case 1 - single winner

translation 1	translation 1	translation 1

ord(sysA, sysB) = +1
ord(sysA, sysC) = +1
ord(sysB, sysC) = +1

↓

???

Case 1 - single winner

translation 1 | translation 1 | translation 1

wins(sysA) = 2
wins(sysB) = 1
wins(sysC) = 0

↓

system with most
+1 rankings wins

???

Case 1 - single winner

<u>translation 1</u> translation 1 translation 1

↓

translation 1

Case 2 - multiple winners

translation 1 | translation 1 | translation 1

ord(sysA, sysB) = +1
ord(sysA, sysC) = -1
ord(sysB, sysC) = +1

↓

no single-best system

???

Case 2 - multiple winners

translation 1	translation 1	translation 1

wins(sysA) = 1
wins(sysB) = 1
wins(sysC) = 1

↓

ord(X,Y) definition guarantees winner

???

Case 2 - multiple winners

translation 1 | translation 1 | translation 1

wins(sysA) = 1
wins(sysB) = 1
wins(sysC) = 1

↓

???

except in case of "circular" results

Case 2 - multiple winners

translation 1 | translation 1 | translation 1

↓

fallback to using best system from training

translation 1

Experiments

Setup

- Participation in ML4HMT-12 shared task
- Submission for Spanish→English; however, our approach is language independent and should also work for Chinese→English
- Systems: `n=4` but has already been used for `n>10`

SVM Optimisation

- We used libSVM for training, 5-fold cross validation to optimise parameters C and γ.
- Experimented with 1) linear, 2) polynomial, and 3) sigmoid kernel setups.
- We ended up using a sigmoid kernel ($C = 2$, $\gamma = 0.015625$) and observed a prediction rate of 68.9608% on the training instances.

Results

Automatic Metrics

- Promising results wrt. small set of features
- Spanish→English
  - Meteor score: 0.323 • Best score observed!
  - NIST score: 7.283 • For some reason *very* bad
  - BLEU score: 0.257 • Not optimised for BLEU

System Contribution

- Another interesting aspect wrt. our approach
- Compare expected and actual contribution
- Strong preference: Moses SMT + Lucy RBMT
- Classifier able to make use of good translations from systems performing bad on corpus level

Conclusion

Findings

- Defined a total order on translation output
- Joint, binarised feature vectors for comparison
- Algorithm for sentence-based combination
- Successfully applied our Machine Learning framework to the ML4HMT-12 shared task

Questions?

Acknowledgements

This work has been funded under the Seventh Framework Programme for Research and Technological Development of the European Commission (grant agreement no.: 249119) through the T4ME contract.