Using Log-linear Models for Tuning Machine Translation Output

Michael Carl

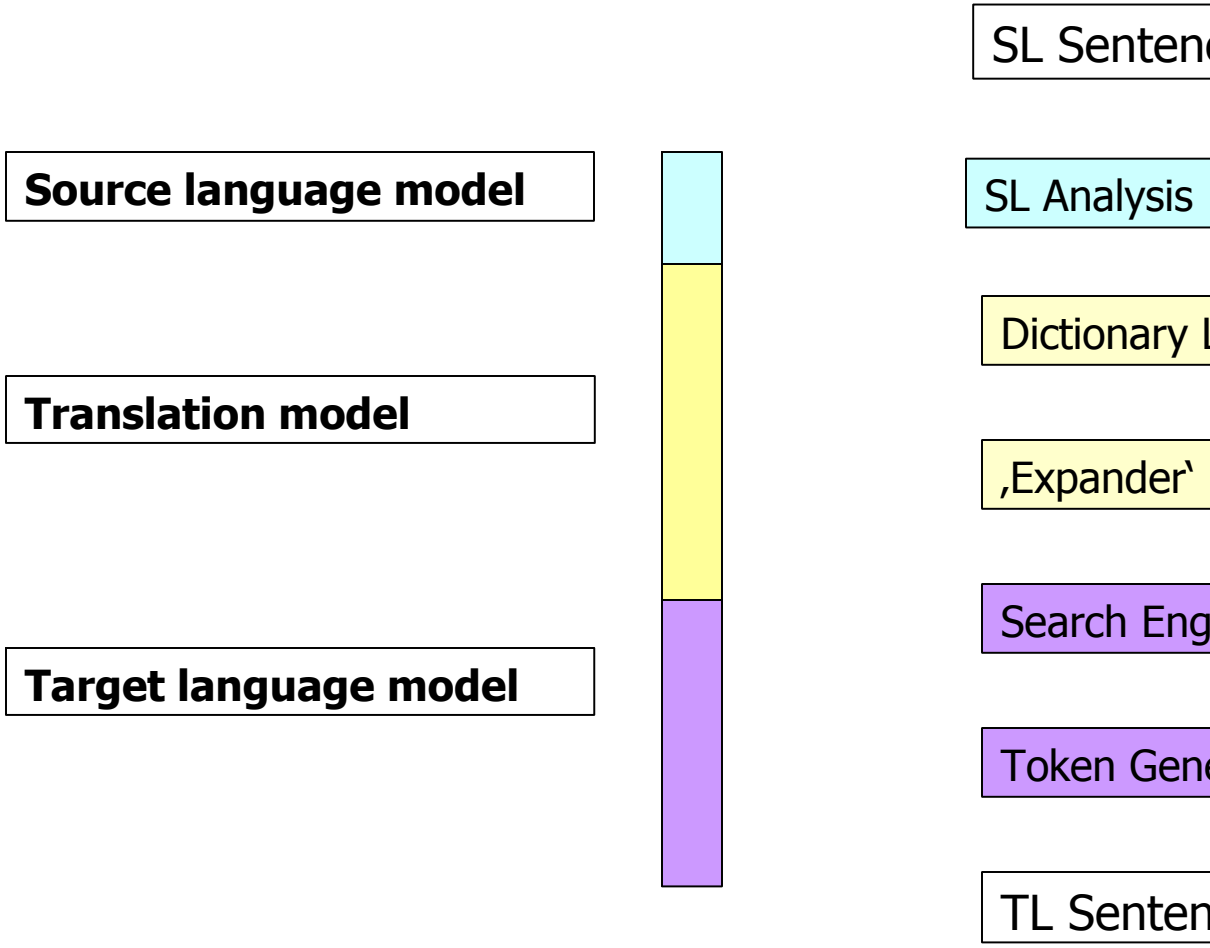
IAI

Overview:

- METIS: Architecture described in session p28 (Friday, 14:40)
  Statistical MT using:
  - Shallow linguistic ressources (SL Analysis, mapping, re-ordering)
  - Hand-made dictionaries (assign weights)
  - Generate (partial) translations and filter
  - Huge TL corpus (n-gram TL models)

- Feature Functions
- Evaluation test set and results
- Conclusion: best results: lemmatisation, tagging, lexical weights

Overview of the System



AND/OR Graph for

SL: Hans kommt nicht

{lu=Hans,c=noun, wnr=1, ...}
@{c=noun}@{lu=hans,c=NP0}. .
,{lu=nicht,c=adv,wnr=3, ...}
@{c=verb}@{lu=do,c=VDZ},{lu=not,c=XX0}.
, {c=adv}@{lu=not,c=XX0}..
,{lu=kommen,c=verb,wnr=2, ...}
@{c=verb}@{lu=come,c=VVB;VVZ}.
, {c=verb}@{lu=come,c=VVB;VVZ},{lu=along,c=AVP}.
, {c=verb}@{lu=come,c=VVB;VVZ},{lu=off,c=AVP}.
, {c=verb}@{lu=come,c=VVB;VVZ},{lu=up,c=AVP}..
.

Types of Feature Functions

- Source features:
  - probabilities of dependencies in SL representations (parse tree dictionary matching)
- Channel features:
  - SL-to-TL alignment and lexical translation probabilities
    - lexical translation weights
- Target features:
  - probabilities of TL sentence (*n-gram* language models)
    - *n*-gram token, lemma, tag models
    - lemma-tag co-occurrence weights

Log-linear feature functions

- Set of specified features *h* that describe properties of the data
- Associated set of learned weights *w* that determine the contribution of each feature.

$$\hat{e} = argmax \sum_m w_m h_m()$$

- Find weights to allow a search procedure (***argmax***) to find the target sentence ***ê*** with the highest probability

Lexical Feature Function

Train ***L(g => e)*** on 10.000 aligned EURPARL sentences:

$$L(g \Rightarrow e) = h(g \Leftrightarrow e) / \sum_e h(g \Leftrightarrow e) + n(g \Rightarrow e)$$

- noise: $n(g \Rightarrow e)$
  g in SL no realization of ***e*** in the TL side
- hit : $h(g \Leftrightarrow e)$
  g in SL and ***e*** in the TL side

Lemma-Tag Cooccurrance Weights

T(lem, tag) = C(lem, tag) +1 / NL + C(lem)

- ***NL:*** number of different CLAWS5 tags (~ 70)
- ***C(lem):***
  number of occurrences of *lem* in the BNC
- ***C(lem,tag):***
  number of co-occurrences of a *lem* and a *tag*

Statistical Language Models

SRILM toolkit:

- *n-gram* language models based on BNC
  - 20K, 100K, 1M and 2M sentences
- Lemma n-gram language models
  - *n={3,4,5}*
- Tag m-gram lanhguage models:
  - *m={3,4,5,6,7}*

Two Evaluation Test Sets
German ==> English

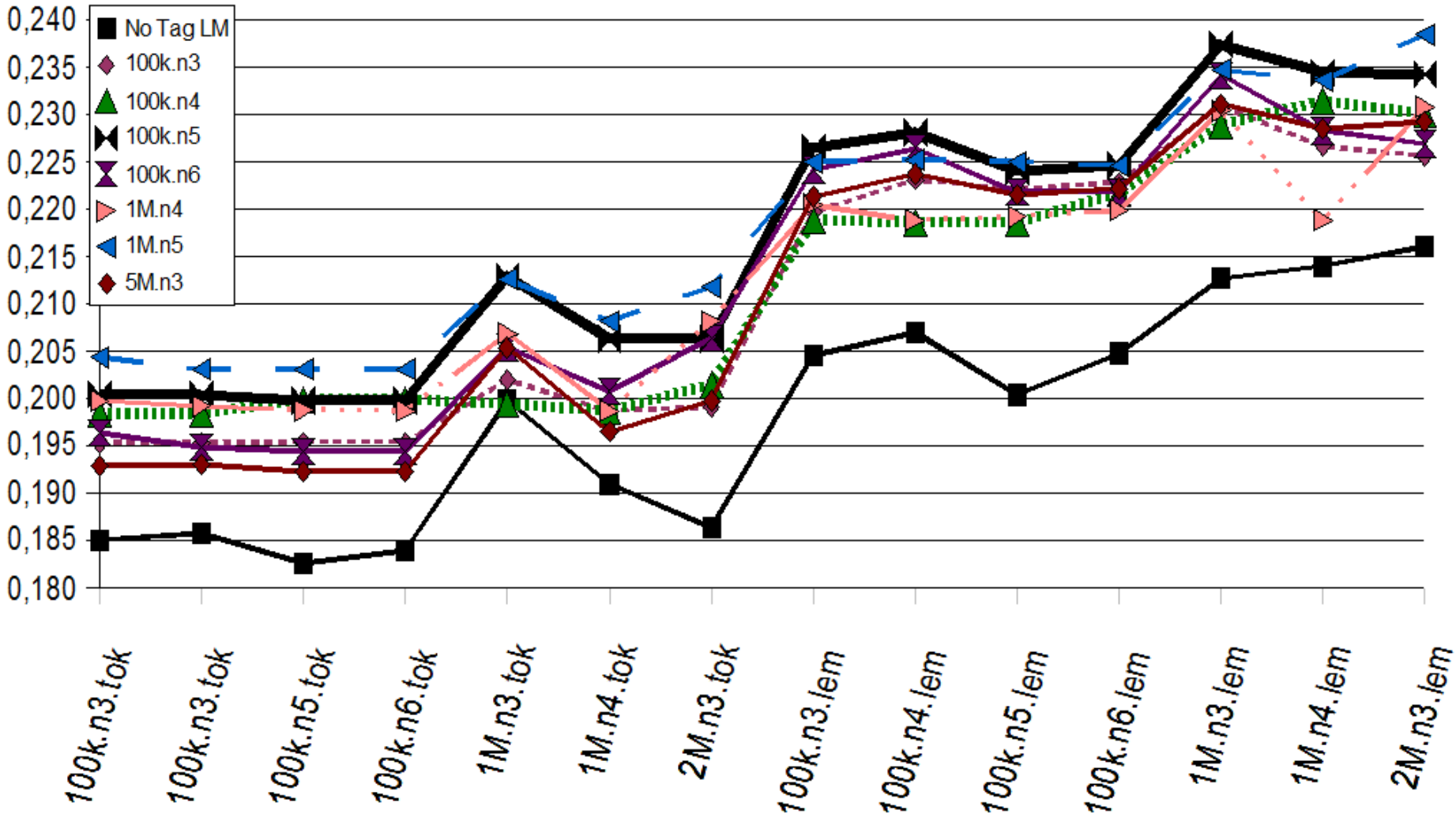
- Tested on a 200 sentences test corpus.
  - lexical translation problems:
    - separable prefixes, fixed verb constructions, degree of adjectives and adverbs, lexical ambiguities, and others
  - syntactic translation problems:
    - pronominalization, determination, word order, different complementation, relative clauses, tense/aspect, etc ..

- 200 sentences selected from the EUROPARL Corpus (extracted from the STAT-MT Website)
  - between 2 and 32 words length (each language side)

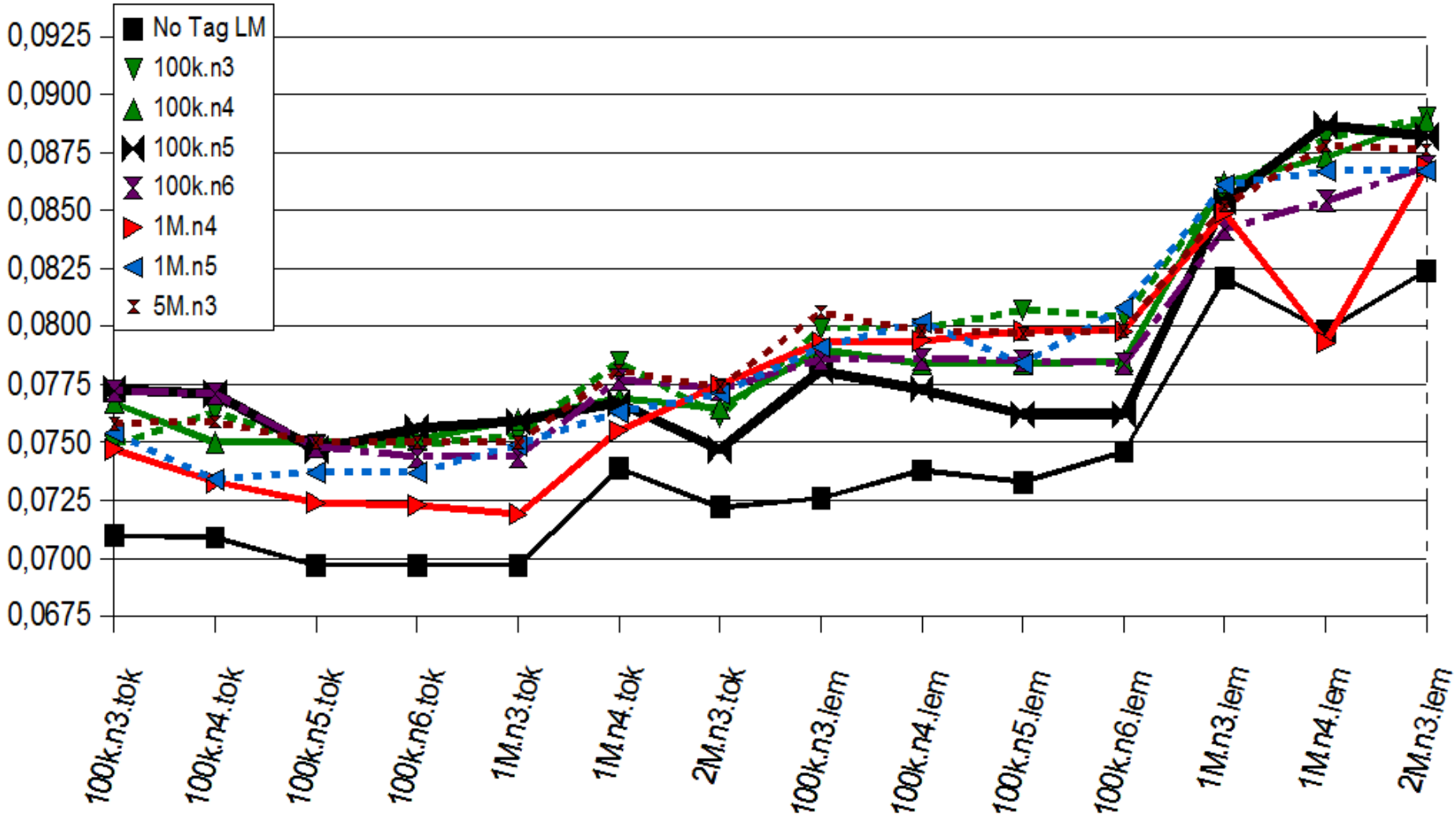
Evaluation

- Start with one feature function (*n-gram* lemma/token model)
- incrementally added feature functions
  - *n-gram* CLAWS5 tag model
  - *m-gram* lemma model
  - Lemma-tag co-occurrence weights
  - Lexical translation weights
- Experimentally assign weights
- Evaluate (with BLEU)

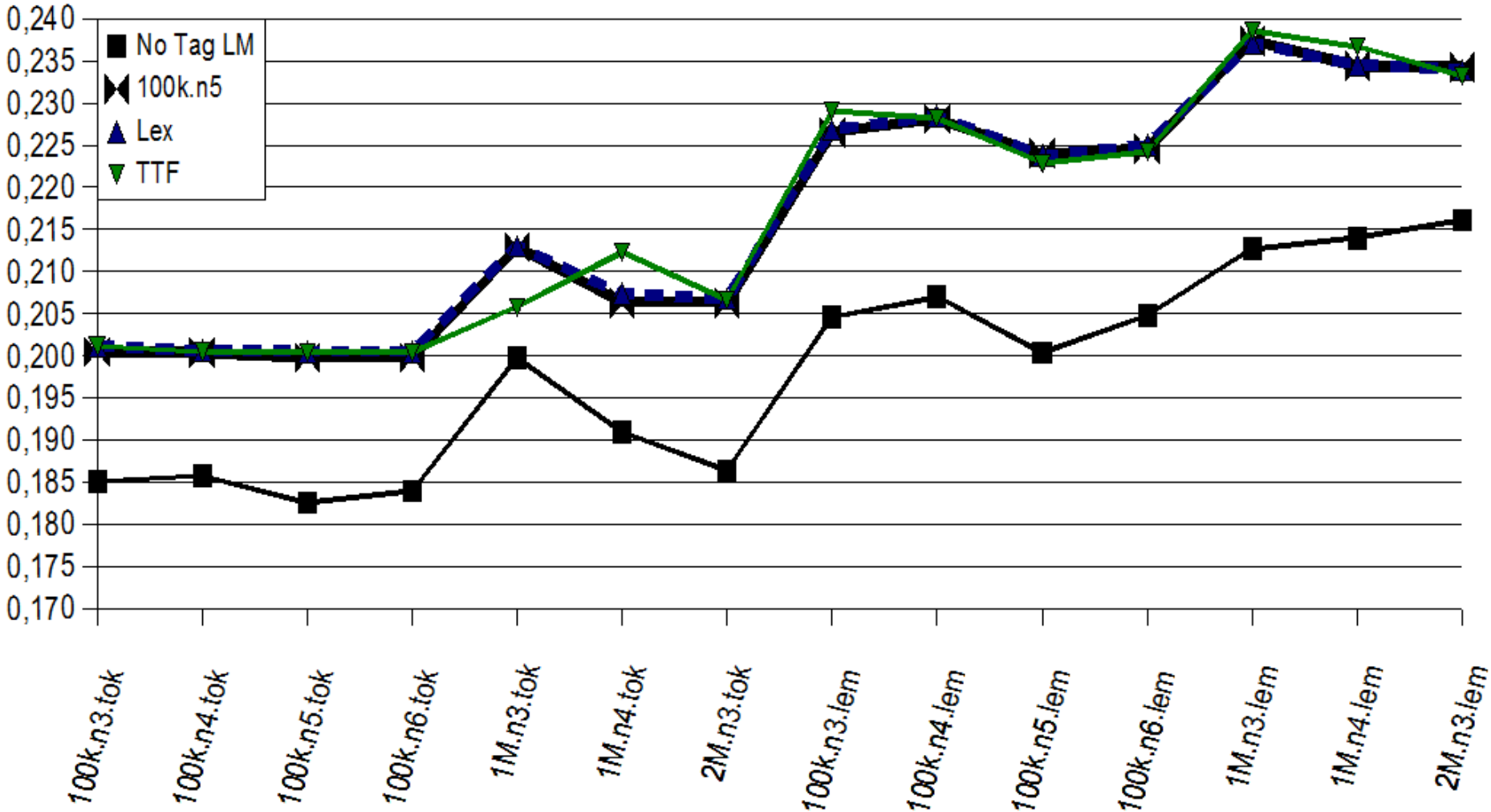
BLEU Evaluation of 200 Test Sentences using token, lemma and tag language models



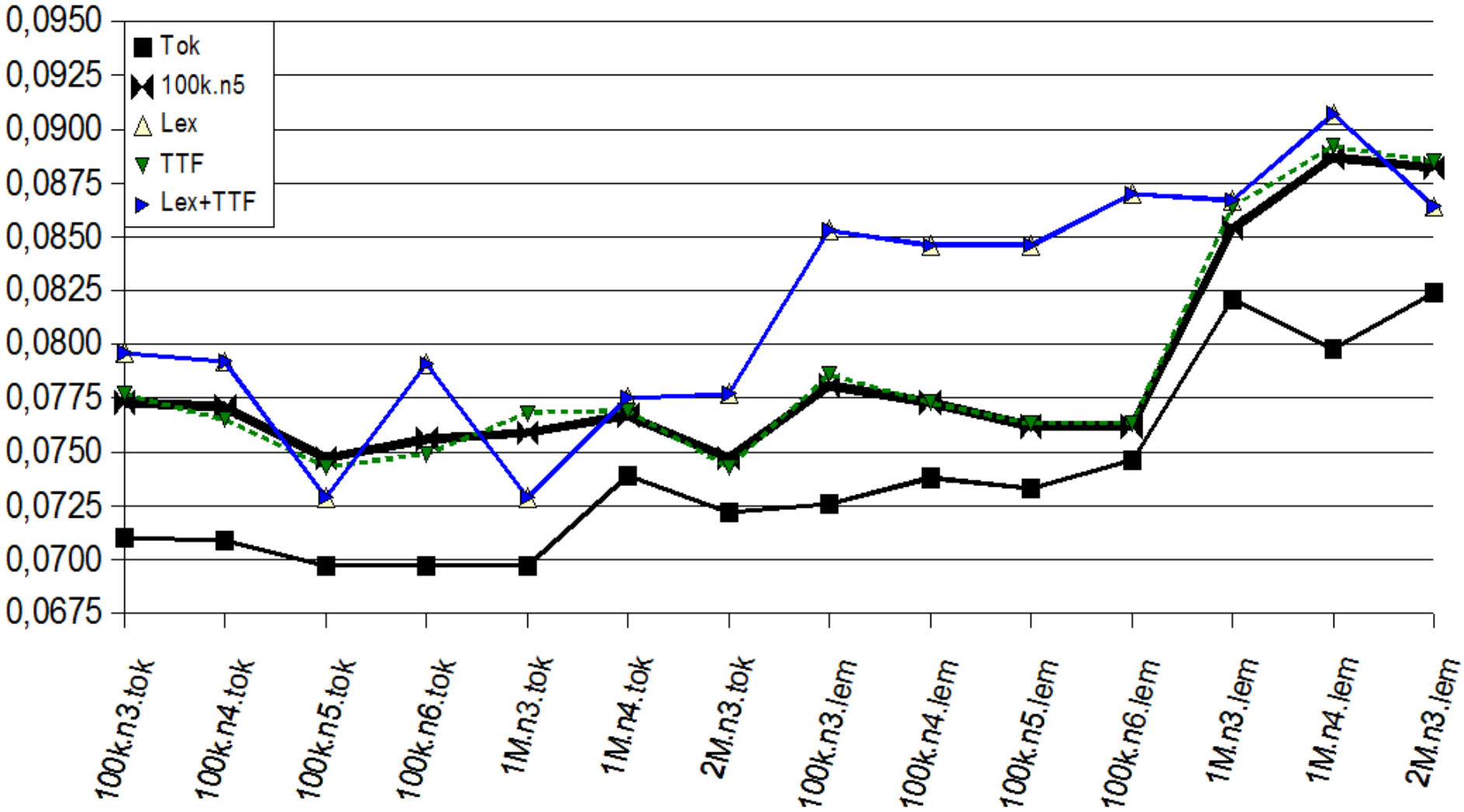
BLEU Evaluation of 200 EUROPARL Sentences using token, lemma and tag language models



BLEU Evaluation of 200 Test Sentences with added lexical (Lex) and token-tag cooccurrence (TTF) models



BLEU Evaluation of 200 EUROPARL Sentences with added lexical (Lex) and token-tag cooccurrence (TTF) models



Conclusion

- Lemma-based models are better than token-based models:
  - increasing size of the training material for lemma models provides better results than increasing the length of the *n-gram* models
- Adding a tag model improves the output in any case:
  - larger values of *n* (in our case *n=5*) may be an easier way to increase perform than to increase the size of the training set
- Token-tag cooccurrance feature function does not help
- Lexical weights are suitable if the training material is similar to the texts to be translated