Compositional Morphology for Word Representations and Language Modelling

Jan Botha and Phil Blunsom
Department of Computer Science, University of Oxford

UNIVERSITY OF OXFORD — DEPARTMENT OF COMPUTER SCIENCE

CLBL++ Model Definition

The log bilinear model (LBL) assigns n-gram probabilities using distributed feature vectors for words and smooth scoring functions.

We extend the standard LBL in two ways:

1. compose word vectors from morpheme vectors (LBL++; see (5))
2. partition vocabulary into word classes for fast normalisation (CLBL)

Predict next representation $\mathbf{p}$ given preceding word vectors $\tilde{\mathbf{q}}_j$:

$$\mathbf{p} = \sum_{j=1}^{n-1} \tilde{\mathbf{q}}_j C_j \quad (1)$$

Score next word w and its class c (word vector $\tilde{\mathbf{r}}_w$, class vector $\mathbf{s}_c$):

$$\nu(w) = \mathbf{p} \cdot \tilde{\mathbf{r}}_w + b_w \quad (2)$$

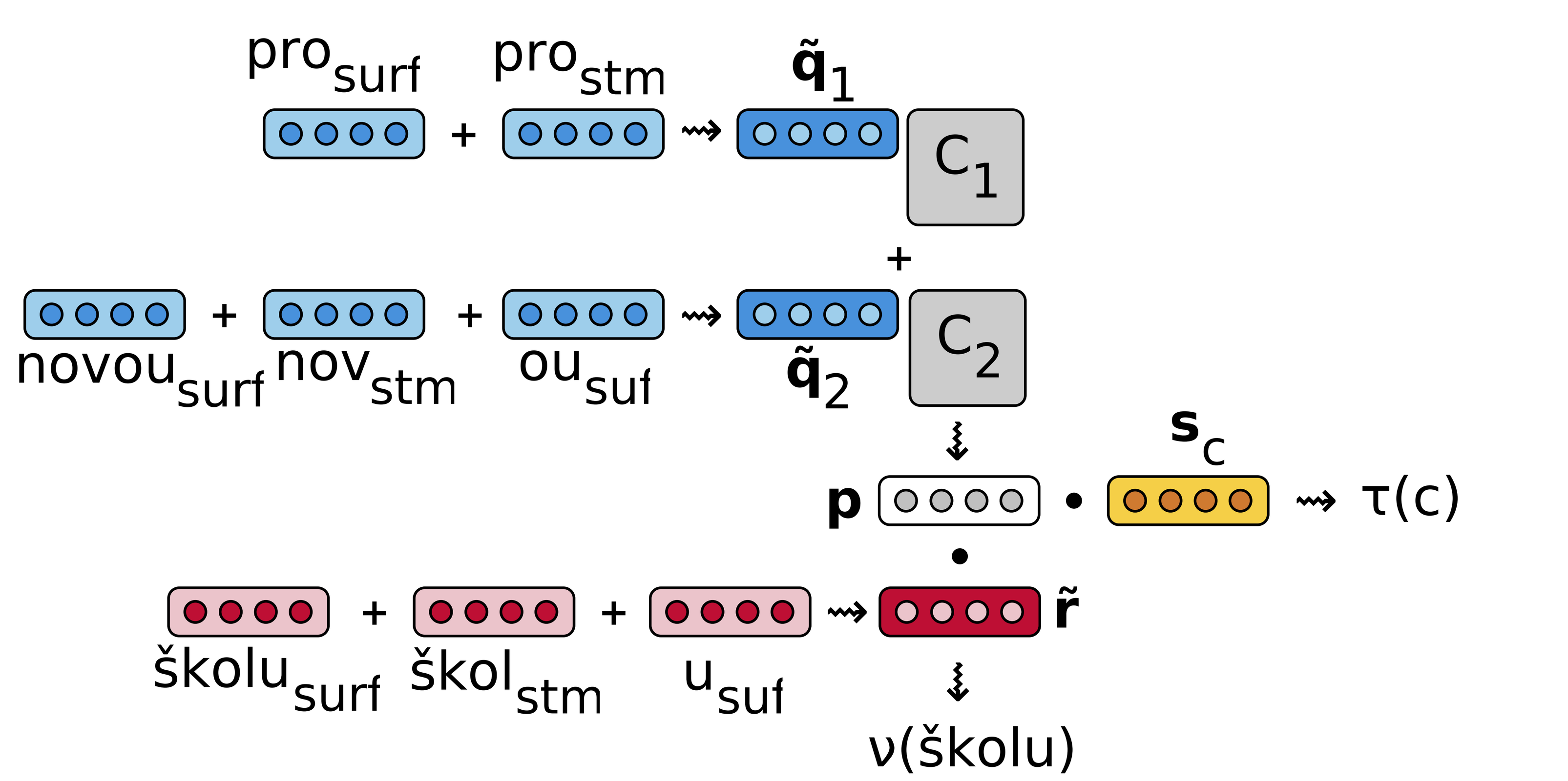
$$\tau(c) = \mathbf{p} \cdot \mathbf{s}_c + t_c \quad (3)$$

Compute probability of word w under model:

$$P(w \mid w_{i-n+1}^{i-1}) = \overbrace{P(class(w) \mid w_{i-n+1}^{i-1})}^{\frac{\exp(\tau(c))}{\sum_{c' \in \{\text{classes}\}} \exp(\tau(c'))}} \overbrace{P(w \mid w_{i-n+1}^{i-1}, class(w))}^{\frac{\exp(\nu(w))}{\sum_{v' \in \{\text{words in class } c\}} \exp(\nu(v'))}} \quad (4)$$

We train the model against an L_2-regularised maximum likelihood objective function using adaptive gradient descent.

Model Diagram



CLBL++ model illustrated for the Czech trigram 'pro novou školu'.

Addition as Composition

1. Map each word type v to a sequence of surface-level factors $\mu(v)$.
2. Define word type vector $\tilde{\mathbf{r}}_v$ as sum of its factor vectors:

$$\tilde{\mathbf{r}}_v \equiv \sum_{f \in \mu(v)} \mathbf{r}_f \qquad (\text{notation: } \overrightarrow{\texttt{word}} \equiv \overrightarrow{factor_1} + \overrightarrow{factor_2} + \ldots) \quad (5)$$

Effect is to tie words with shared morphemes, e.g.

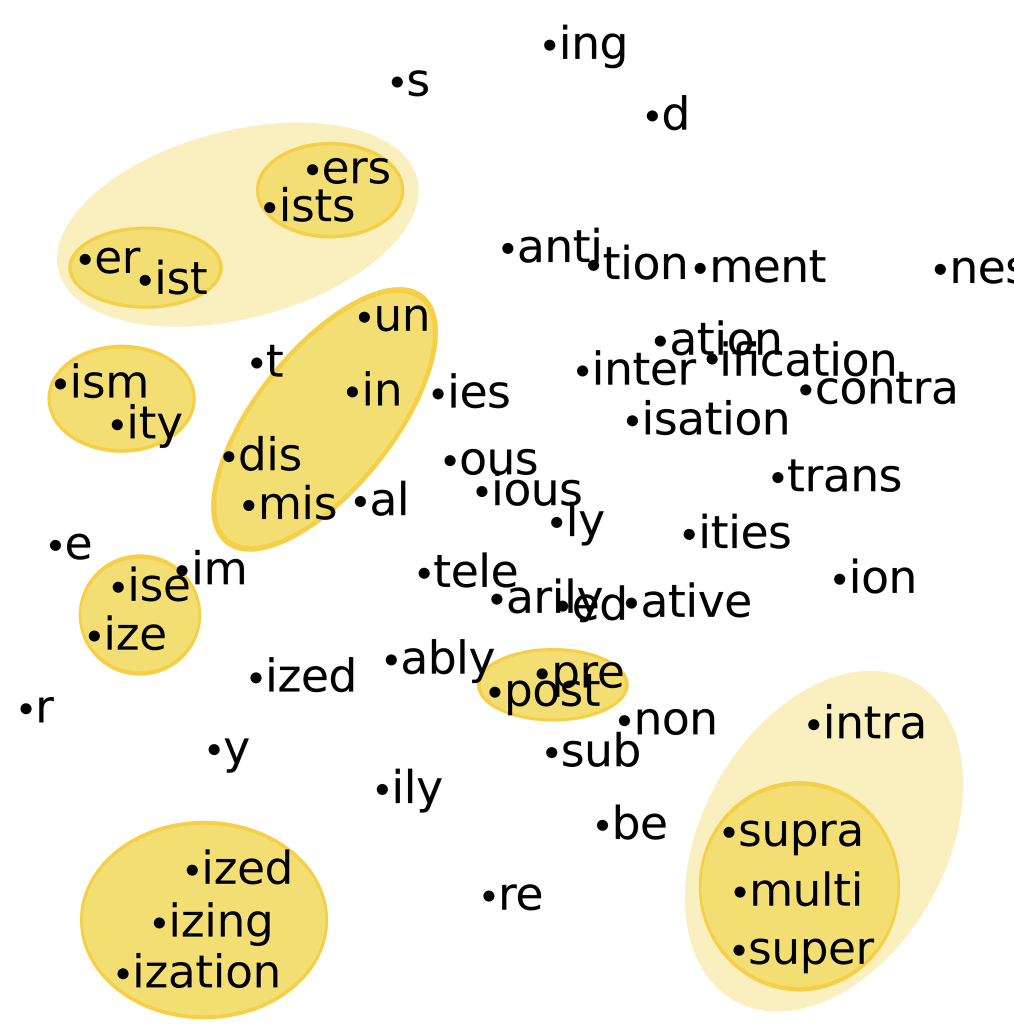
$$\overrightarrow{\texttt{unexpected}} = \overrightarrow{unexpected}_{surf} + \overrightarrow{un}_{pre} + \overrightarrow{expect}_{stm} + \overrightarrow{ed}_{suf}$$

$$\overrightarrow{\texttt{expectations}} = \overrightarrow{expectations}_{surf} + \overrightarrow{expect}_{stm} + \overrightarrow{ation}_{suf} + \overrightarrow{s}_{suf}$$

Including surface form as factor means our approach

- appreciates order: $\overrightarrow{\texttt{hangover}} \neq \overrightarrow{\texttt{overhang}}$
- handles non-compositionality: $\overrightarrow{\texttt{greenhouse}} \neq \overrightarrow{green} + \overrightarrow{house}$
- overcomes noise from unsupervised segmentor (we used Morfessor).

Morpheme Embeddings Acquired



t-SNE projections of English affix vectors learnt by CLBL++, with shading added manually for emphasis.

Software available via http://www.clg.ox.ac.uk/resources

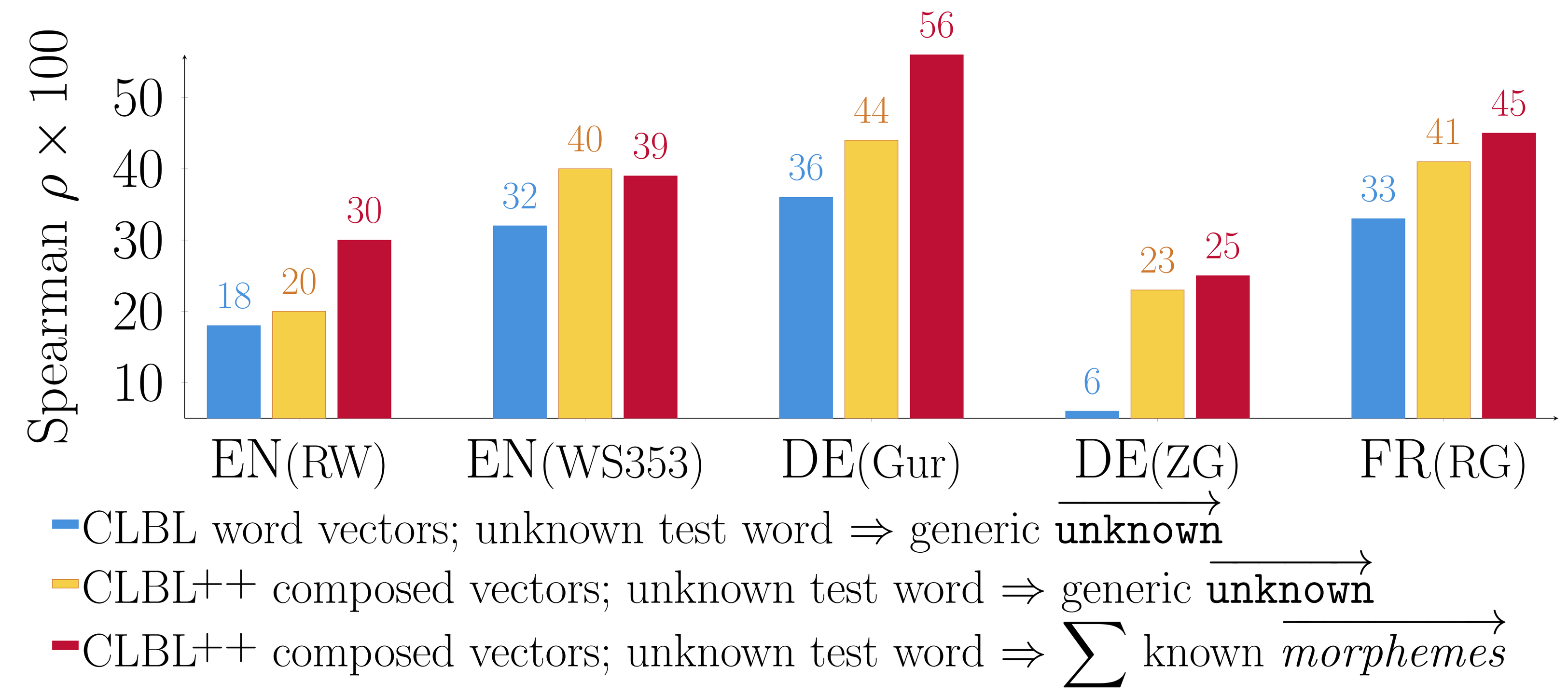
Summary

Integrate morphology into vector-based language models

- Simple, scaleable, unsupervised method
- Learns morpheme vectors as part of model
- Word classes enable integration into MT decoder
- Improvements in three evaluation settings and multiple languages

Word-pair Similarity Rating Task

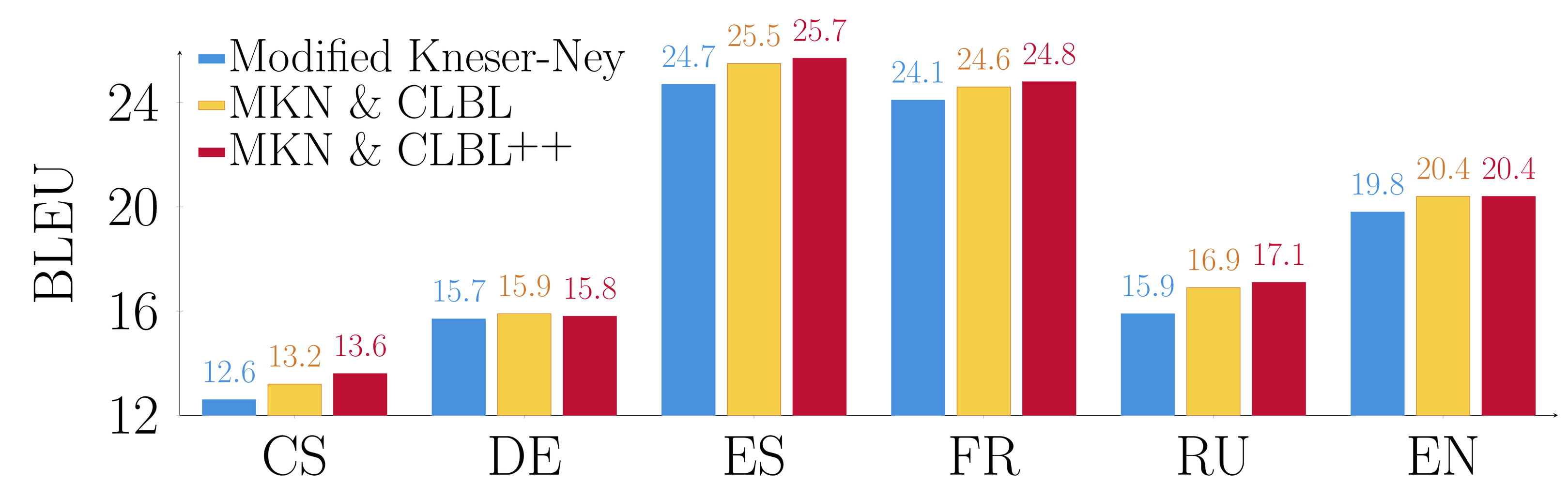
The CLBL++ model learns vectors that obtain stronger correlation with human judgements of word-pair similarity. Morpheme vectors allow more subtle handling of unknown words.



Results for measuring similarity of word-pairs using learnt vectors.

Machine Translation Task

Class-based partitioning of vocabulary speeds up computation of normalised language model probabilities. This is crucial for the large vocabularies of morphologically rich languages, and enabled integration of the CLBL/CLBL++ inside an MT decoder.



Results for translation into different languages, varying the LMs.