Cross-lingual supervised text classification

Hauke Licht

Cologne Center for Comparative Politics
hauke.licht@wiso.uni-koeln.de

Tutorial prepared for COMPTEXT 2022

May 4, 2022

Background

Text-as-data

Goal

- study political behavior and communication using quantitative methods
- ultimately, answer substantively interesting research questions about politics

Premise

- text generation is political $\rightsquigarrow$ text is important artifact of political behavior
- texts indicates their authors' political preferences, attitudes and beliefs, and strategies

General approach

- extract "features" (data) from text suitable for quantitative analysis
- requires reducing complexity of human language and numeric representation of text

Cross-lingual quantitative text analysis

Study political behavior and communication across languages

- multilingual institutional contexts
- enable application of QTA methods in *comparative* research (Lucas et al. 2015)

The central challenge

> *develop an alignment between the conceptual representations of the model across languages so that we know a particular scaling, topic, or class in one language is comparable with the representation in another language.* (Lucas et al. 2015, 259)

$\rightsquigarrow$ obtain **identical measurements** for documents that indicate the **same concept**

Cross-lingual quantitative text analysis

The central challenge

obtain **identical measurements** for documents that indicate the **same concept**

$\rightsquigarrow$ even if they are written **in different languages**

Why this is challenging

Tower of Babel problem (Chan et al. 2020, Maier et al. 2021)

different vocabularies, words do not co-occur, similar words have dissimilar contexts

$\rightsquigarrow$ makes language-independent inference hard

Cross-lingual measurement equivalence

the substance of political discourses varies across polito-linguistic context

$\rightsquigarrow$ might necessitate to account for context

Cross-lingual supervised text classification

Text-as-data approaches and tasks

Manual content analysis

- text classification (coding/categorization)
- text scaling (through pairwise comparison)

Quantitative text analysis

Text classification

- dictionary analysis
- **supervised text classification**
- topic modeling

Text scaling

- unsupervised (*Wordfish*, *Wordshoal*)
- semi-supervised (*Wordscores*, LSA)

Cross-lingual supervised text classification

Text classification

Assign each document in a corpus to a one of several pre-defined categories

Supervised text classification

"Learn" how to assign documents to categories based on a (small) subset of documents for which you know which categories they belong to

How? apply supervised machine learning techniques

Cross-lingual supervised text classification

"Learn" to assign documents to categories when they are written in different languages

Today's running example

Lehmann and Zobel (2018) data set

- corpus of human-coded election manifestos
- quasi-sentences coded into **issue categories**: immigration, integration, "others"
- manifestos of parties from 14 countries $\rightsquigarrow$ **8 languages**

$\rightsquigarrow$ view the dataset **online**

Our goal

discriminate between the **immigration/integration** and the "others" category

Relevance

- study competition on the immigration/integration issue
- learn to extrapolate (expensive) human codings
  - to new countries (cross-lingual transfer)
  - to new domain (à la Osnabrügge, Ash and Morelli 2021)

Approaches

Separate analysis

1. split multilingual corpus by language
2. train separate, language-specific classifiers
3. apply them to classify unlabeled documents

$\leadsto$ pool resulting measurements for cross-lingual comparison

Input alignment ← our focus today

1. transfer documents' representations to a common denominator (i.e., "align" them)
2. train a single, cross-lingual classifier
3. apply it to classify unlabeled documents

$\leadsto$ use resulting measurements in cross-lingual comparison

Input alignment

Approaches

Machine translation

Translate all documents to a single **target language**

Multilingual embedding

Represent all documents in a joint, multilingual **embedding space**

A unifying idea

- transfers documents to a common denominator
- enable their joint analysis within a single model
  - direct cross-lingual comparison
  - information-sharing across languages
  - resource-efficient

Input alignment approaches

Machine translation

Idea

- overcome Babel problem by translating all documents to one target language
- the target language is the common denominator

Evidence

works quite very well for typical tasks

- topic modeling (Lucas et al. 2015; de Vries, Schoonvelde and Schumacher 2018; Reber 2019)
- dictionary analysis (Windsor, Cupit and Windsor 2019)
- supervised classification (Courtney et al. 2020)
- textual similarity (Düpont ant Rachuj 2021)

Implementation (I)

Machine translation

Human translators too expensive. Instead, rely on state-of-the-art **neural machine translation** (NMT) methods.

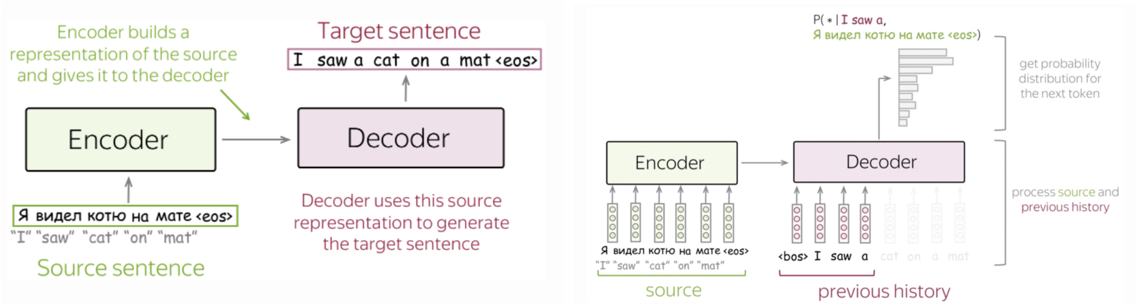


Figure 1: Translation as seq-to-seq problem. *Source:* Lena Voita's "NLP Course" (**online**)

Implementation (II)

Machine translation

Human translators too expensive. Instead, rely on state-of-the-art **neural machine translation** (NMT) methods.

Using commercial service

- state-of-the-art NMT technology (e.g., *Google Translate* or *DeepL*)
- extensively evaluated (see literature cited on last slide)
- "black box" (cf. Chan et al. 2020)
- quite expensive

Use open-source NMT model ← our approach today

- open-source $\rightsquigarrow$ reproducible and free of charge
- massively pre-trained (e.g., Facebook research's M2M)

Free machine translation

pip install easynmt

The easyNMT python package provides a simple interface to download and use several large pre-trained NMT models:

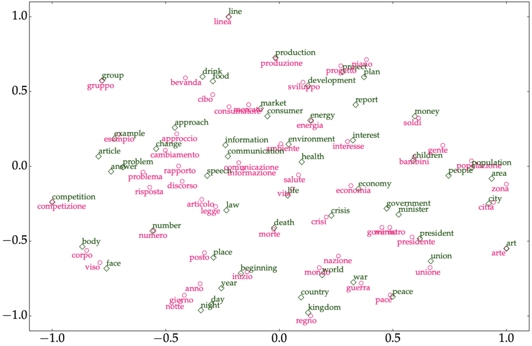
```
from easynmt import easyNMT

# download and instantiate a pre-trained M2M model
model = easyNMT("m2m_100_418M")

# translate a single sentence
model.translate("Guten Tag liebe Freunde!", target_lang="en")
```

⇝ see **this** Colab Notebook for an illustration

Multilingual embedding

Idea

align documents by representing them in a multilingual embedding space



Evidence

applied in existing contributions for

- topic modeling (Chan et al. 2020)
- text scaling (Glavas, Nanni and Ponzetto 2017b; Goist 2021)
- textual semantic similarity (Radford, Dai and Golder 2021)
- supervised classification (Glavas, Nanni and Ponzetto 2017a; Dai and Radford 2019; Licht 2022)

Approaches

Multilingual *word* embedding

- many different approaches (see Ruder, Vulić and Søgaard 2019)
- currently most common among ME-based contributions (e.g., Chan et al. 2020, Goist 2021)

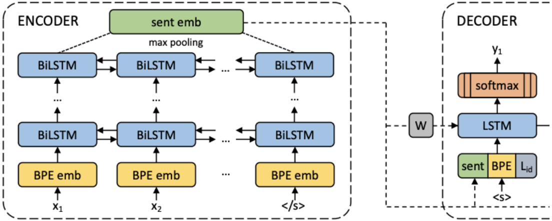
Multilingual *sentence* embedding

- again different approaches (next slide)
- well-suited for applications when documents have sentence-like lengths
  $\rightsquigarrow$ **supervised text classification** (Licht 2022)

Multilingual sentence embedding (I)

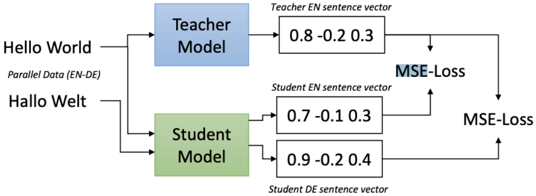
LASER

use (fixed-size) embedding produced by encoder in NMT (Artexte and Schwenk 2019)



Knowledge distillation

extend pre-trained monolingual sentence embedding model to new languages (Reimers and Gurevych 2020)



Multilingual sentence embedding (II)

pip install sentence-transformers

The sentence-transformers python package provides a simple interface to download and use several large pre-trained MSE models:

```
from sentence_transformers import SentenceTransformer

# download and instantiate a knowledge-distilled XLM-R model
model = SentenceTransformer("paraphrase-xlm-r-multilingual-v1")

# translate a single sentence
model.embed("Guten Tag liebe Freunde!")
```

⇝ see **this** Colab Notebook for an illustration

Application in Cross-lingual supervised text classification

Supervised classification: recap (I)

Terminology

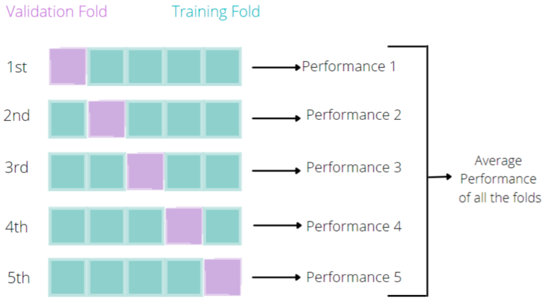
- **classes**: the set of outcome categories
- **label**: the class a document belongs to
- **sample**: a single document and its label
- **data set**: a collection of samples

```
   | label | text                                 |
   |-------|--------------------------------------|
-> |   +   | "Our government is doing a great job!"       | <-- a sample
   |   -   | "Those crooks in power are super corrupt!" |
   |  ...  |                                      |
```

Supervised classification: recap (II)

Training & Evaluation

- **training**: fit a model to a data set to optimize its classification accuracy
- **held-out samples**: samples not used to train a classifier
- **evaluation**: see how a classifier performs in held-out samples
- ***k*-fold cross validation** (CV): train and evaluate a classifier on *k* partitions of a data set



Supervised classification: recap (I)

Procedure

1. select (a set of) classification models to try
2. sample documents in corpus into training and test data sets
3. (repeatedly) sample documents into k folds and create k train–val CV splits
4. cross-validate $\rightsquigarrow$ find the best-performing model or the best hyper parameter values for a model

Alternative

omit CV and use validation data set instead

Cross-lingual supervised text classification

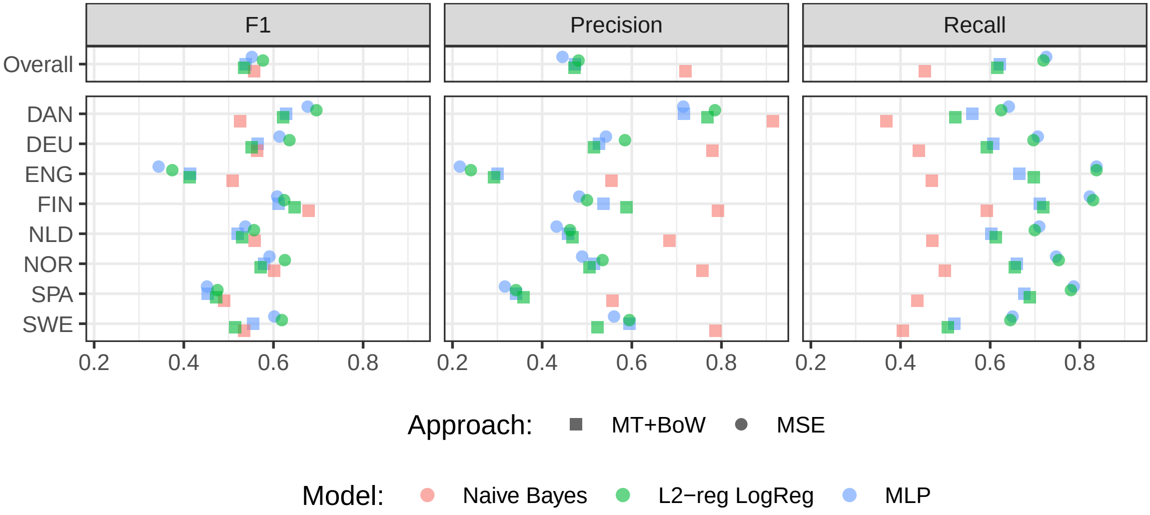
MT approach

1. machine translate all documents into the target language
2. train classifier on documents' text representations in the target language
3. evaluate classifier on held-out documents' target language versions

MSE approach

1. embed all documents using a pre-trained MSE model
2. train classifier on documents' embeddings
3. evaluate classifier on held-out documents' target language versions

Results


F1
Precision
Recall
Overall
DAN
DEU
ENG
FIN
NLD
NOR
SPA
SWE
0.2
0.4
0.6
0.8
Approach: MT+BoW MSE
Model: Naive Bayes L2−reg LogReg MLP

Resources

Links to Google Colab notebooks

- inspect the Lehmann and Zobel (2019) data set (**link**)
- *input alignment*
  - how to machine-translate with the `easyNMT` package (**link**)
  - how to sentence-embed with the `sentence-transformers` package (**link**)
- *supervised classification*
  - sample the train, CV, and test indices (**link**)
  - train MT+BoW classifiers (**link**)
  - train MSE-based classifiers (**link**)

Data

- the cleaned Lehmann and Zobel data set (incl. machine-translated texts, **link**)
  - XLM-R sentence embeddings (**link**)
- the train, CV, and test indeces configuration JSON (**link**)