Exercise 06 – Token Level Transfer

Chair XII for Natural Language Processing

Prof. Dr. Goran Glavaš
Fabian David Schmidt
Benedikt Ebing

Datasets

WikiANN – How was it sourced?

- 282 languages sourced from Wikipedia:
  1. Get name mentions from Wikipedia in English and classify them (location, person, organization) (“silver-standard”)
     - E.g., en/Mitt_Romney -> PER
  2. Propagate the classifications through cross-lingual Wikipedia links
     - en/Michigan | Location → uk/Мічиган
  3. Train language specific name tagger on the instances resulting from step 1 and 2 in a self-training framework
     - Additionally, consider commonness of entities and topic relatedness
  4. Cross-Lingual Entity Linking (if name cannot be resolved)
     - Linking identified entities back to English via word level translations

WikiANN – General Issues

- Few entities with a lot of references (due to design choices)
- Many instances aren´t proper sentences
- High number of single mention instance

WikiANN – Cross-Lingual Transfer Issues

- „Silver-standard“ data
  - E.g., misalignment in labels (e.g. „If I were a boy“ labelled as organization)
  - Span issues
  - Entities that are not names
- Primarily used version is a (stratified) down-sampled version of the original (additionally limited to predefined sizes)
  - Changing the label distribution
  - Discarding a lot of instances
  - Discarding languages (that do not have enough certain entity types)
  - Estimation of “real-world” performance is limited

MasakhaNER – How was it sourced?

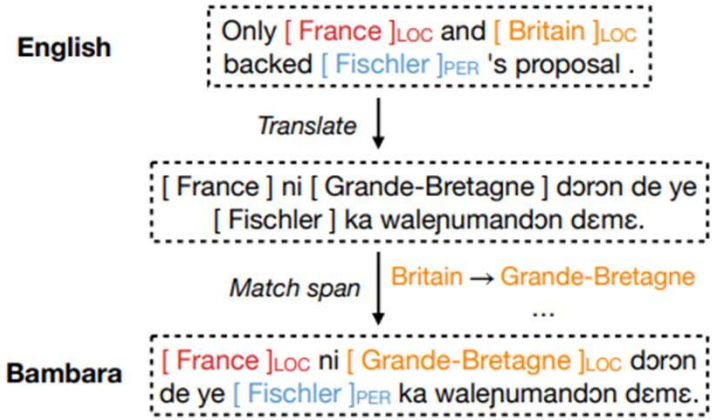
- 10 languages obtained from local news sources
- Annotated by native speakers that are from the same regions as the news sources and volunteered
- Entity types: person, location, organization, data & time
- High inter annotator agreement

MasakhaNER – Why is it valuable?

- Covers languages that are underrepresented in NLP for pre-training and fine-tuning
- Sizeable number of human annotated instances (even small training sets)
  - No translations!
- High density of mentions in each instance

Translate-Train for Token-Level Transfer

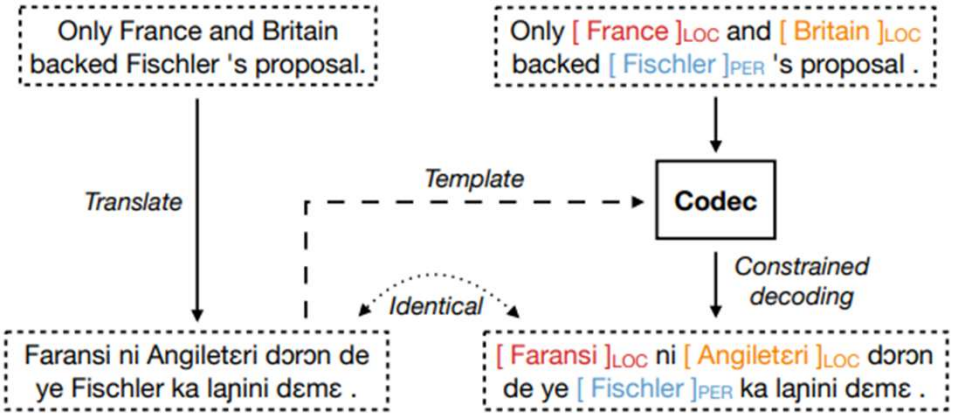
Translate-Train: EasyProject



Label Projection (depending on marker):

- XML-like markers allow for simple mapping entities (e.g., <a> in the source sentence gets mapped to <a> in the translated target sentence)
- Square brackets „[]“ more difficult:
  - Sequential
  - Fuzzy String Matching (by translating only the Entity and comparing against the marked span in the translated sentence)
- Post-processing in case of destroyed markers (“[protest] ing or [[protesting]”)

Translate-Train: Codec



(b) CODEC: Translate w/o markers then insert markers with constrained decoding

Label Projection:

- Complex method to ensure that the markers are inserted at a meaningful position
  - Pruning unlikely opening markers (e.g., likelihood difference between generating an opening marker and not generating an opening marker)
  - Pruning unlikely search branches during decoding
  - Reranking of hypothesis
  - Merging of multi-label spans (e.g., treat each label as a separate instance)

Translate-Train: Word Aligner

- Get similarity matrix: $S = h_x h_y^T$
- Softmax across rows- and columns of S
- Get the alignment matrix by intersection:

$$A = (S_{xy} > c) * \left(S_{yx}^T > c\right)$$

  - where c is a threshold and $A_{ij} = 1$ means the tokens at position i and j are aligned
- Label Projection:
  - Naively applying the mappings (many errors)
  - More sophisticated approaches:
    - Filtering the mappings
    - Mapping span-wise (e.g., start and end of a labeled span)

Discuss the approaches – Marker-based

- EasyProject:
  - Type of marker
  - Matching strategy in multi-label scenarios
  - Introduce noise to the text (i.e., markers)
  - Not applicable to translate-test
- Codec:
  - Hyperparameters for the methods: pruning thresholds, number of generated candidates, merging of multi-label instances
- General:
  - Ability of the machine translation model to retain/predict markers
    - Off-the-shelf open-source models often fail to retain the markers
      - Fine-tuning helps (e.g., train translation on marked instances)
    - Commercial translation systems (Google Translate, …) are very robust and retain markers
  - Robust post-processing to clean noisy markers (“[[Germany]”)

Discuss the approaches – Word Alignment

- No additional noise in the translation (compared to EasyProjection)
- Depends on the quality of the word aligner
  - Word alignment remains a difficult task
    - E.g., translations that do not allow for a word-to-word alignment
  - “Good” word aligners do exist for high-resource languages, but more critical for low-resource languages
  - More recently: Collecting parallel data and using sentence transformers (next lectures/exercise)
- Depends on pre-tokenization (before applying the alignment) and exact mapping algorithm (particularly in the case of translate-test)

Discussion:
Do you think AwesomeAlign is a fair baseline to Codec and EasyProject?

Discourse: Translate-Test for Sequence Level Tasks

How to improve translate-test?

- Translate-Test performance sensitive to translation quality
  - ➔Better translation models lead to larger performance gains
- Recent improvements primarily for English centric models
- Overcoming the distribution shift at inference time
  - Train on noisy roundtrip translated training data (English->Target Language->English)

Discussion:
Is the comparison between translation-based transfer and zero-shot transfer fair?
Should we focus on better English backbone models and translation?

Additional Exercises

Estimate all parameters of an IBM Model 2 (all q_p, q_w) on the given parallel corpus with MLE (using the alignments). Compute the probability your model assigns to the three translations given in the corpus (given the alignment).

Source	Target	Alignment (Target,Source)
the dog	Hund der	(1,2),(2,1)
the dog	der Hund	(1,1),(2,2)
the dog	die Hund	(1,1)(2,2)

Recap: IBM Model 2

- Alignment $a_k = (i, j)$ – means that t_i is aligned to s_j

$$P(\mathbf{t}|\mathbf{s}) = \prod_{i=1}^{m} q_p(j|i,m,n) * q_w(t_i|s_j)$$

Position alignment score (for positions i and j given lengths m and n)

Word translation scores (regardless of positions of words)

$$q_p(j|i,m,n) = \frac{c(j|i,m,n)}{c(i,m,n)}$$

$$q_w(t|s) = \frac{c(t,s)}{c(s)}$$

t ➔ Target sentence
t_i ➔ i-th token of t
s ➔ Source sentence
s_j ➔ j-th token of s

Solution: q_w

- $q_w(der|the) = \frac{2}{3}$
- $q_w(die|the) = \frac{1}{3}$
- $q_w(Hund|dog) = \frac{3}{3}$
- All other $q_w = 0$

Source	Target	Alignment (Target,Source)
the dog	Hund der	(1,2),(2,1)
the dog	der Hund	(1,1),(2,2)
the dog	die Hund	(1,1)(2,2)

Solution: q_p

- $q_p(2|1,3,2) = \frac{1}{3}$
- $q_p(1|2,3,2) = \frac{1}{3}$
- $q_p(1|1,3,2) = \frac{2}{3}$
- $q_p(2|2,3,2) = \frac{2}{3}$
- All other $q_p = 0$

Source	Target	Alignment (Target,Source)
the dog	Hund der	(1,2),(2,1)
the dog	der Hund	(1,1),(2,2)
the dog	die Hund	(1,1),(2,2)

Compute the probability your model assigns to the three translations given in the corpus

- $P(t^1|s^1)$

$$= q_p(2|1,3,2) * q_w(Hund|dog) * q_p(1|2,3,2) * q_w(der|the)$$

$$= \frac{1}{3} * 1 * \frac{1}{3} * \frac{2}{3} = \frac{2}{27}$$

- …